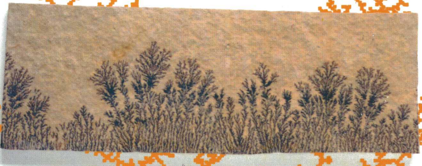
HANS LAUWERIER

COMPUTER



SIMULATIES

DE WERELD ALS MODEL

ARAMITH UITGEVERS

B

Computersimulaties

Hans Lauwerier

Computersimulaties

De wereld als model

Aramith Uitgevers Bloemendaal

CIP-gegevens Koninklijke Bibliotheek, Den Haag

Lauwerier, Hans

Computersimulaties. De wereld als model. Hans Lauwerier
Bloemendaal, Aramith
Met illustraties
ISBN 90 6834 106 5
NUGI 855
Trefwoord: computersimulatie

© 1992 Hans Lauwerier
© 1992 licentie-uitgave Aramith Uitgevers, onderdeel van Uitgeverij J.H. Gottmer/H.J.W. Becht bv, Postbus 160, 2060 AD Bloemendaal
Omslagontwerp: Hard Werken Design
Typografische vormgeving: Studio Cursief, Amsterdam
Druk: Ten Brink Meppel
Alle illustraties zijn vervaardigd door de auteur

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen, of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor zover het maken van kopieën uit deze uitgave is toegestaan op grond van artikel 16b j° het Besluit van 20 juni 1974, St.b. 351, gewijzigd bij Besluit van 23 augustus 1985, St.b. 471 en artikel 17 Auteurswet 1912, dienen de daarvoor wettelijk verschuldigde vergoedingen te worden voldaan aan de Stichting Reprorecht (Postbus 882, 1180 AW Amstelveen).
Voor het overnemen van gedeelten uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich tot de uitgever te wenden.

Inhoud

Inleiding 7
1. Pythagoras en priemgetallen 11
2. Figuren van Lissajous 18
3. Wandelkaart en borduurpatroon 23
4. Zwevingen en moiré 31
5. Potentialen en velden 40
6. Zelfgelijkvormige plantengroei 49
7. Wiskunde en wiskunst 57
8. Het pad van de schildpad 64
9. Lijnen die een vlak vullen 75
10. Van woord tot plant en boom 89
11. Kartografie, het bedrog van landkaarten 95
12. Tussen de sterren 103
13. Maak uw eigen planetarium 110
14. Relaxatietrillingen 116
15. Het hart als elektrisch systeem 125
16. Orde en chaos 136

De computerprogramma's 147
Bibliografie 167

Inleiding

De computer is niet meer weg te denken uit onze samenleving. Veel huisgezinnen beschikken over een of meer exemplaren. Vaak dient de computer als moderne schrijfmachine met alle voordelen van tekstverwerking. Talloze softwarepakketten stellen de gebruiker in staat privé-administratie uit te voeren. Wie buiten kantoortijd thuis verder wil gaan behoeft niet meer met zware tassen te zeulen, maar steekt slechts enkele diskettes in zijn binnenzak. De jongere generatie houdt zich ongetwijfeld bezig met computerspelletjes, hoewel ook ouderen dit vermaak wel zullen kennen.

Dit boek richt zich echter tot een andere groep van gebruikers, tot al degenen die geïnteresseerd zijn in wetenschappelijk onderzoek, die nieuwsgierig zijn te weten hoe de wereld in elkaar zit, trouwe lezers van de wetenschappelijke rubrieken van kranten en tijdschriften.

Het boek is voortgekomen uit een serie maandelijkse bijdragen van de schrijver aan het tijdschrift *Natuur en Techniek* onder de titel 'Simulatica'. In elke bijdrage wordt in kort bestek een onderwerp uit de natuurwetenschappen, natuurkunde, biologie, chemie, astronomie enzovoort behandeld en wordt getoond hoe aan de hand van een wiskundig computermodel inzicht kan worden verworven. Voor dit boek is uit het beschikbare materiaal een keuze gedaan op grond van de volgende criteria: het onderwerp moet interessant zijn of een zekere mate van actualiteit bevatten, en het onderwerp mag geen grote eisen stellen aan de wiskundige kennis van de lezer. Aldus is een reeks van zestien hoofdstukken ontstaan waarin iedereen wel iets van zijn gading zal vinden.

Om de bedoeling van het boek duidelijk te maken gaan we even in op het vijftiende en voorlaatste hoofdstuk. Daarin wordt de werking van het menselijk hart besproken, het hart als elektrisch systeem. Een eenvoudig computermodel laat iets zien van de stabiliteit van het kloppende hart. Een volgend model laat zien hoe we de werking van het hart kunnen verstoren met een klein elektrisch stootje. Begrijpelijk wordt dan hoe het kan gebeuren dat een kleine elektrische stimulus van een bepaalde grootte en toegediend op een bepaald tijdstip het hartritme op fatale wijze kan ontregelen. Bij een hart kunnen verschillende ritmestoornissen optreden. In het wiskundige model zijn er twee onderling gekoppelde klokken die niet even snel lopen. Daarbij kan een zogenaamde fasekoppeling optreden waarbij de tweede klok q-maal rondgaat wanneer de eerste klok p-maal rondgaat. De combinatie $p = 4$ en $q = 3$ correspondeert dan met een bekend ziektebeeld, het fenomeen van Wenckebach. Een aantal computerexperimenten heeft betrekking op de wijze van fasekoppeling. Als interessant bijprodukt wordt dan tevens duidelijk hoe een pacemaker werkt.

Andere computerexperimenten laten zien wat er gebeurt wanneer men een aantal grootheden laat variëren, de mate van koppeling en de intensi-

teit van de stimulus bijvoorbeeld. Dat is een vorm van wetenschappelijk onderzoek, een simulatie van wat er op wetenschappelijk gebied in de grote laboratoria aan de universiteiten gebeurt.

Wetenschappelijk onderzoek aan de hand van wiskundige modellen is nu ook thuis mogelijk, de moderne computer verschaft ons zoveel mogelijkheden, dat deze uitspraak heel serieus genomen mag worden. Dit boek opent de weg daartoe. Vele modellen waarvan de grote pioniers op het gebied van de moderne natuurwetenschappen zich bediend hebben, zijn – in soms wat vereenvoudigde vorm – bruikbaar materiaal voor thuiswerk. Er gaat dan ook een grote charme vanuit om hun experimenten met eigen middelen uit te kunnen voeren.

Van de lezer wordt behalve enige geringe wiskundige kennis en belangstelling voor de bètawetenschappen alleen wat handigheid in het omgaan met een computer gevraagd. De programma's zijn geschreven voor een MS-DOS-computer in een moderne vorm van Basic, een computertaal die nauw verwant is aan bijvoorbeeld Pascal of C, maar die minder formeel is en minder administratieve rompslomp kent. Bovendien is de moderne versie van Basic beter leesbaar en komt de wiskundige kern daarvan beter tot zijn recht. Met een compiler als Turbo Basic, Power Basic of Quick Basic kan ook de volle computersnelheid benut worden. Overigens draaien veel programma's ook goed met het bij MS-DOS 5.0 behorende QBasic.

Gestreefd is naar veelzijdigheid, zodat er voor ieder wel wat te vinden is. Computerkunst of ruimtevaart, textielpatronen en moiré of plantengroei, kartografie of het meten van chaos. Een groot aantal computerprogramma's stelt de lezer in staat zich actief met het onderwerp te bemoeien, en bevat simulaties om beter vertrouwd te raken met het onderwerp, tot beter inzicht te komen, als een vorm van wetenschappelijke activiteit.

De hoofdstukken zijn op een enigszins losse manier geordend. Wie van computerkunst houdt zou zich in principe kunnen beperken tot de hoofdstukken 3, 4 en 7. Maar ook in de hoofdstukken 5 en 6 kan hij materiaal van zijn gading vinden. De hoofdstukken 2, 5 en 14 zijn typisch natuurkundig, maar hoofdstuk 14 over relaxatietrillingen preludeert weer op het al eerder genoemde hoofdstuk 15 over de ritmiek van het hart. Veel aandacht is besteed aan de algoritmische groei van planten, de zogenaamde Lindenmayer-systemen. Programma's werkend volgens de schildpadmethode of 'turtlegraphics' en het principe van backtracking kunnen fraaie groeisels op het computerscherm doen ontstaan.

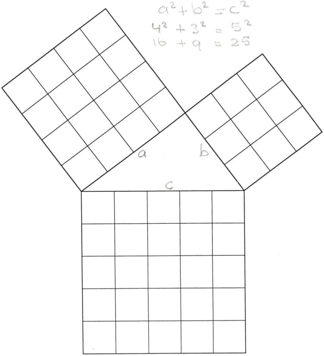
De hoofdstukken 11, 12 en 13 vallen onder de gemeenschappelijke noemer van aardrijkskunde of sterrenkunde. De moderne atlassen laten ons steeds mooiere kaarten zien, maar helaas geven ze vaak een vertrokken beeld van de werkelijkheid. In de wereldkaart van Mercator zijn gebieden op hogere breedtegraden sterk vergroot weergegeven en lijkt Groenland van vergelijkbare grootte te zijn als heel Afrika. Er zijn wel andere projectiemethoden om grote gebieden van het aardoppervlak af te beelden, maar geen enkele methode is in staat een zuiver beeld te geven. Het is namelijk een wiskundig gegeven dat een bol niet zonder vervorming tot een plat vlak kan worden gemaakt. Met een computerexperiment kan men zien wat een bepaalde projectiemethode waard is. Wat in werkelijkheid op het aardoppervlak een vierkant is, blijkt op de landkaart

soms ernstig vervormd te zijn. In hoofdstuk 13 worden een paar simulaties besproken waarin een planeet of een komeet om de zon draait. Daarbij kan een deel van het planetenstelsel op het beeldscherm zichtbaar worden gemaakt, waarbij de planeten natuurgetrouw om de zon bewegen. Af en toe schiet er dan een komeet voorbij. Het is verrassend te zien hoe de onderlinge stand van de planeten zich telkens wijzigt wanneer het programma aan de wetten van Kepler gehoorzaamt. In hoofdstuk 12 is een simulatie gegeven van een ruimtevaartuig dat passief tussen twee om elkaar heen draaiende sterren beweegt. Het blijkt mogelijk te zijn dat de satelliet daarbij het stelsel enige malen bezoekt en daarna door de zwaartekracht van een der sterren de ruimte wordt ingeslingerd. De beweging is buitengewoon gevoelig voor kleine veranderingen, zoals de onderzoeker in zijn simulatieprogramma kan ondervinden. Het is een klassiek voorbeeld van een instabiel of chaotisch dynamisch gedrag. Meer over chaos treft de lezer aan in het laatste hoofdstuk.

Chaos is een intrigerend probleem, een begrip dat in de mode schijnt te zijn en waarover ook veel onzin is gezegd. Hier laten we zien dat er aan het begrip chaos van een eenvoudig systeem heel wat vastzit, dat er verschillende vormen van chaos zijn, en dat chaos in vele gevallen meetbaar is. Ondanks de beperking tot een paar eenvoudige modellen als het kneden van deeg en het Verhulst-model van geremde groei is dit hoofdstuk toch een van de langste geworden, maar misschien ook een van de meest interessante. De bespreking van het zogenaamde gemengde Verhulstmodel sluit dan ook heel nauw aan bij recent wetenschappelijk onderzoek.

1 Pythagoras en priemgetallen

Over de historische figuur Pythagoras is maar weinig bekend. Het meeste dat we van hem weten is wat veel latere auteurs in de klassieke Oudheid ons weten te vertellen, en dat zijn vaak sterke verhalen. Vaststaat dat hij een veelzijdig geleerde was, en dat hij zich in het bijzonder bezighield met de eigenschappen van de natuurlijke getallen, 1, 2, 3, 4, 5, ... De naar hem genoemde stelling over vierkanten beschreven op de zijden van een rechthoekige driehoek heeft hij kennelijk ontleend aan de wiskunde van de Mesopotamiërs. In bewaard gebleven kleitabletten in spijkerschrift vinden we daar bewijzen van. De stelling van Pythagoras is meetkundig geïllustreerd in figuur 1.1.



figuur 1.1 De stelling van Pythagoras

Zoals we kunnen constateren is de som van de oppervlakken van de vierkanten beschreven op de rechthoekszijden gelijk aan het oppervlak van het vierkant op de schuine zijde, door de Grieken de hypotenusa genoemd. In de figuur is een leuk speciaal geval afgebeeld. De zijden van de kleinste twee vierkanten zijn respectievelijk drie en vier eenheden. De zijde van het grote vierkant is dan precies vijf eenheden, omdat $5 \star 5 = 3 \star 3 + 4 \star 4$.

Men heeft wel eens gedacht dat de Egyptenaren de stelling van Pythagoras gebruikten om bij de bouw van hun piramiden en graftomben over een zuivere rechte hoek te kunnen beschikken. Men nam dan eenvoudig een lang koord en verdeelde dit met knopen in twaalf gelijke stukken. Zogenaamde koordspanners vormden dan hiermee de zijden van een rechthoekige driehoek zoals in figuur 3.1. Helaas is dit toch een moderne fantasie gebleken; het zou gekund hebben, maar in de literatuur van de Oudheid zijn er geen aanwijzingen voor te vinden. Voor ons is het misschien nog bruikbaar bij de aanleg van een tuin, als een zogenaamde tuinmansconstructie.

Een dergelijk stel getallen a, b en c waarvoor geldt dat

$$a^2 + b^2 = c^2$$

pleegt men tegenwoordig een *Pythagorees tripel* te noemen. Andere voorbeelden zijn 5,12,13 en 6,8,10. Dat laatste tripel is niet echt nieuw, omdat het onmiddellijk uit 3,4,5 door verdubbeling kan worden afgeleid.

Een eenvoudig recept

De vraag rijst hoe we meer van die tripels kunnen ontdekken anders dan door domweg, al of niet met de computer, te proberen. Welnu, daar bestaat een eenvoudig wiskundig recept voor. We nemen twee willekeurige natuurlijke getallen m en n, en daarmee berekenen we

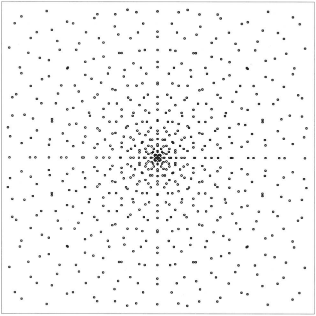
$$a = m^2 - n^2, \qquad b = 2mn, \qquad c = m^2 + n^2$$

We laten het aan de lezer over om na te rekenen dat het inderdaad klopt. De keuze $m = 2$, $n = 1$ geeft al meteen het eerste voorbeeld. Voor $m = 3$, $n = 2$ vinden we het tripel 5,12,13.

Om een overzicht te krijgen van de mogelijkheden kunnen we alle combinaties a,b die met c een Pythagorees tripel vormen als een punt in een x,y-diagram weergeven. Men zou ook ruitjespapier kunnen gebruiken en het met bijvoorbeeld 3,4 corresponderende ruitje zwart maken. Vanwege de symmetrie is het dan aardig om daarnaast ook alle combinaties met negatieve getallen te markeren. Met 3,4 zijn dat er in totaal acht stuks, te weten $\pm3,\pm4$ en $\pm4,\pm3$. We kunnen aldus een fraai tafelkleedje vervaardigen, met kruissteekjes bijvoorbeeld (zie figuur 1.2). Met behulp van de computer kunnen we er ons snel een voorstelling van maken. We bedienen ons daarbij van het programma PYTKLEED.

In het kleedje kunnen we merkwaardige regelmatige patronen ontwaren. Punten lijken zich langs gebogen lijnen te ordenen. Uit de definitie

figuur 1.2
Het tafelkleed van Pythagoras



van x en y kunnen we afleiden dat dit kegelsneden moeten zijn, krommen die eveneens tot het cultuurgoed van de oude Grieken behoren.

Priemgetallen

Priemgetallen hebben ons altijd gefascineerd. Meer dan tweeduizend jaar geleden wisten de Grieken al dat er oneindig veel waren, en nu is het vinden van een heel groot priemgetal een uitdaging voor geavanceerde computers. De eerste priemgetallen

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, \ldots$$

kunnen we zo opschrijven. Handige rekenaars zien onmiddellijk dat die getallen geen delers hebben wanneer we tenminste 1 en het getal zelf uitsluiten. Maar hoe kunnen we uitmaken of een groot getal als 1234567 deelbaar of niet deelbaar is? Er bestaat daarvoor helaas geen goede methode anders dan gewoon proberen of het getal soms gedeeld kan worden door een kleiner getal d. Voor het vaststellen van deelbaarheid door 2, 3, 5 en 11 bestaan heel eenvoudige trucjes, maar daarna wordt het moeilijk. Er moet dus almaar stug gerekend worden en dat kunnen we wel aan de computer overlaten. Is het getal n deelbaar door d dan is ook n/d een deler. Is n toevallig een kwadraat, dan kunnen die getallen aan elkaar gelijk

zijn, maar anders is altijd een van die twee kleiner dan $\sqrt{n}$. Bij het zoeken naar een mogelijke deler hoeven we dus niet verder te gaan dan de grens $\sqrt{n}$. Bij het zoeken naar een eventuele deler d van n zouden we ons kunnen beperken tot een deler die zelf een priemgetal is. Helaas zit er weinig systematiek in, en is het moeilijk daar een eenvoudig computerprogramma op te baseren. We doen daarom maar liever wat meer, en soms overbodig, werk door alle oneven getallen 3, 5, 7, 9, 11, 13, 15, 17, ... als mogelijke deler te proberen. In de praktijk blijkt dat heel goed te werken en kunnen we met het programma PRIEM snel vaststellen of n priem is of niet. Het getal 1234567 blijkt dan geen priemgetal te zijn, maar wel het produkt van de twee priemfactoren 127 en 9721.

Computerproblemen

Wie dit programma of een variant daarvan op zijn eigen computer wil uitvoeren kan op een paar moeilijkheden stuiten. Intern werkt de computer doorgaans met registers van twee bytes, dat wil zeggen zestien bits. Zo'n register biedt plaats aan $2^{16} = 65536$ gehele getallen. Met een ±-teken erbij correspondeert dat met de getallen -32768 tot 32767. Dat zijn de gewone 'integers', waar elke standaardcomputer mee werkt. Berekeningen met uitkomsten buiten dat interval leiden tot ontsporingen omdat de computer intern alle 'integers' tot binnen de genoemde grenzen afknot. Gelukkig bieden de meeste computertalen wel de mogelijkheid tot zogenaamde 'long integers', maar die moeten dan wel aan het begin van het programma als zodanig gedeclareerd worden. Intern reserveert de computer vier bytes voor deze getallen, en de grenzen zijn nu van -2^{31} tot $2^{31}-1$, waarbij $2^{31}-1 = 2\ 147\ 483\ 647$. Dat laatste getal is overigens een beroemdheid, een zogenaamd getal van Mersenne, waarvan Euler in 1772 vaststelde dat het een priemgetal was.

In het programma PRIEM, geschreven voor een moderne Basicversie, is daarvan gebruik gemaakt door n aan het begin als een 'long integer' te declareren. De structuur van het programma is doorzichtig genoeg om snel vast te stellen wat er gebeurt. Eerst worden alle factoren 2 verwijderd en vervolgens de factoren 3. Zodra een deler d van n gevonden is gaan we verder met het onderzoek van het quotiënt n/d. Aldus wordt een getal n met meer delers telkens verkleind en schieten we met de berekening snel op. Vervolgens worden alle volgende gehele getallen 5, 7, 9, 11, ... als mogelijke deler geprobeerd. Zodra een deler gevonden is wordt die afgedrukt en wordt met j het aantal delers bijgehouden.

Om te toetsen of d een deler is van n kunnen we twee methoden gebruiken. De eenvoudigste manier is

```
WHILE N MOD D = 0 ...
```

maar helaas werkt dit niet in sommige Basic-versies, omdat de modulusopdracht slechts op gewone (korte) gehele getallen betrekking heeft. In Q(uick)Basic gaat het wel goed. In het programma PRIEM gebruiken we de INT-opdracht:

```
WHILE N\D = INT(N\D) ...
```

die waarschijnlijk altijd lukt, maar die misschien wat langer duurt.

Delen kost tijd

Met de compilerversie Power Basic, de opvolger van Turbo Basic, is het zelfs mogelijk getallen bestaande uit achttien cijfers te factoriseren. Power Basic beschikt namelijk over de unieke mogelijkheid om intern met integers van viervoudige lengte (quads) te werken. Het grootst toelaatbare gehele getal is dan 2^{63}-1, of in decimale notatie

9 223 372 036 854 775 807

Hetzelfde programma kan gebruikt worden, maar wel moet aan het begin de regel waar N als long integer gedeclareerd wordt vervangen worden door

```
DEFLNG J,F,T : DEFQUD N
```

wat dus wil zeggen dat de getallen J, F, T long integers kunnen zijn en dat N een viervoudige lengte heeft. Bij Power Basic moet men wel bedacht zijn op een klein probleempje. Bij het inlezen van een geheel getal vanaf het toetsenbord zijn maar achttien decimale posities mogelijk. Het grootste getal dat in het hier geboden programma gefactoriseerd kan worden bestaat dus uit achttien negens, toch nog groot genoeg!

Het is aardig eens een paar tijden te vergelijken. We nemen daartoe het eerder genoemde getal 2^{31}-1. Met de door de auteur gebruikte Compaq 386/20 met coprocessor kost het in de moderne interpretatieve Quick-Basic-versie 58 seconden. De compilerversie Quick Basic heeft er achttien seconden voor nodig, terwijl zowel Turbo Basic als Power Basic er tien seconden over doen. Mersenne en Fermat hebben zich in 1643 beziggehouden met de ontbinding van 100 895 598 169. Fermat vond de ontbinding 112303 ⋆ 898423. Met Power Basic gaat het nu in zes seconden. Een recente computer van het i486/33 type presteert hetzelfde in slechts een enkele seconde, een resultaat dat de kracht van de moderne computer demonstreert maar dat ons ook met diep ontzag vervult voor het denk- en schrijfwerk van de twee wiskundigen, meer dan driehonderd jaar geleden.

In een publikatie daagde de auteur en wiskundige Van de Craats zijn lezers uit hun krachten te beproeven op de ontbinding van het getal 31 059 639 803 977. Na 83 seconden meldt ons Power-Basic-programma het resultaat

1597553 ⋆ 19442009

Twee priemfactoren! Met de zojuist genoemde moderne computer gaat het nog eens zesmaal zo snel.

Het tafelkleedje van Van der Pol

De wis- en natuurkundige Balthasar van der Pol (1889-1959), internationaal vooral bekend geworden door zijn onderzoek aan de naar hem genoemde differentiaalvergelijking, was een zeer veelzijdig geleerde met veel hobby's en met een grote experimentele vaardigheid. Een van zijn 'grappen' was de vervaardiging van een 'priemgetallenkleedje'.

Het begrip priemgetal kan in de wiskunde heel ruim opgevat worden. Zodra een nieuwe getalsoort gedefinieerd is en de eigenschappen van deelbaarheid gelden, heeft het zin te spreken van een priemgetal. Een dergelijk getal heeft dan per definitie geen echte delers, dat wil zeggen geen eenheidsgetallen enzovoort. Beschouwt men gehele getallen, dan zijn 1 en -1 de eenheden en is -13 een priemgetal, net als 13.

Getallen van Gauss

Van der Pol maakte gebruik van de zogenaamde gehele getallen van Gauss, getallen die we kunnen opvatten als de roosterpunten van ruitjespapier, of anders gezegd punten $P(x,y)$ in een Cartesisch assenstelsel met gehele coördinaten. De getallen stellen we voor als de combinatie $x + iy$ of $x + yi$, waarbij i de zogenaamde imaginaire eenheid is. Kenners zien meteen dat het gaat om complexe getallen, maar wie daar geen verstand van heeft mist nauwelijks iets. De hoofdzaak is dat we met die getallen, geschreven in de vorm $x + yi$, op de gewone algebraïsche wijze kunnen rekenen, mits we maar i^2 door -1 vervangen wanneer dat mogelijk is. Dus is

$$(a + bi) \star (c + di) = (ac - bd) + (ad + bc)i$$

Die regel van vermenigvuldiging kunnen we interpreteren als het vermenigvuldigen van vectoren. Is O de oorsprong, zijn (a,b) de coördinaten van het punt P en (c,d) die van het punt Q, dan kunnen we zeggen dat de vermenigvuldiging van de vectoren OP en OQ een produktvector OR oplevert waarbij R het punt is met de coördinaten $(ab\text{-}cd,ad\text{+}bc)$. We kunnen constateren dat bij die vermenigvuldiging ook voldaan is aan

$$OP \star OQ = OR$$

wat hetzelfde is als

$$(a^2 + b^2) \star (c^2 + d^2) = (ac - bd)^2 + (ad + bc)^2$$

De uitdrukking $a^2 + b^2$, dus de kwadraatlengte van de vector OP, noemt men de norm van het getal $a + bi$. We nemen nu bijvoorbeeld

$$8 - i = (2 + 3i) \star (1 - 2i)$$

Het getal $8 - i$ blijkt dus opgebouwd te kunnen worden uit twee delers en kan dus zelf deelbaar worden genoemd. Voor de overeenkomstige normen stellen we vast dat $65 = 13 \star 5$.

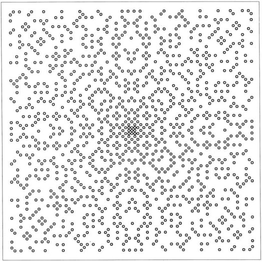
Elk getal van Gauss is deelbaar door ±1 en ±*i*. Die tellen als eenheden en hun norm is 1. Bij de definitie van priemgetallen moeten we ze derhalve uitsluiten. Het getal 3+2*i* is een priemgetal. We stellen even vast dat de norm 13 is. Zou 3+2*i* deelbaar zijn, dan zou ook de norm 13 deelbaar zijn. Die is echter zelf een priemgetal en zo is 3+2*i* een priemgetal van Gauss. Met 3+2*i* zijn ook 3-2*i*, -3+2*i* en -3-2*i* priemgetallen. Maar ook ±2±3*i* zijn priemgetallen, en zo komen alle priemgetallen van Gauss voor in groepen van acht (of vier). De gewone priemgetallen zijn in het stelsel van Gauss niet noodzakelijk priem. Zo is bijvoorbeeld

$$13 = 13 + 0i = (3 + 2i) \star (3 - 2i)$$

Maar wel kan men bewijzen dat alle gewone priemgetallen die gelijk zijn aan een viervoud minus 1 ook als Gauss-getal priem zijn.

Weven van priemen

Van der Pol liet ter gelegenheid van een internationaal congres te Amsterdam in 1954 een kleedje weven waarin de priemgetallen van Gauss binnen een vierkant met hoekpunten ±40±40*i* als blokjes waren afgebeeld (figuur 1.3). Het patroon is kennelijk symmetrisch omdat de priemgetallen in groepjes als ±*a*±*bi* en ±*b*±*ai* voorkomen. De originele exemplaren zijn zeldzaam geworden, maar het patroon kan zonder moeite op het beeldscherm worden verkregen met behulp van het programma GAUSS. Bovendien kan men het patroon op een willekeurige plaats begrenzen.



figuur 1.3
De priemgetallen van Gauss

2 Figuren van Lissajous

Natuurkundige experimenten hebben altijd iets fascinerends, vooral wanneer met eenvoudige hulpmiddelen verrassende resultaten bereikt kunnen worden. Grote indruk op zijn tijdgenoten maakte de Franse fysicus Jean B.L. Foucault, die met zijn in het Parijse Panthéon opgehangen slinger in 1851 de aswenteling van de aarde kon aantonen. Een aantal jaren later (omstreeks 1857) bestudeerde een andere Franse fysicus, Jules-Antoine Lissajous, wat er gebeurt wanneer een slinger tegelijk aan twee verschillende slingerbewegingen deelneemt. Men zou zich kunnen voorstellen dat een Parijse kwajongen de slinger van Foucault een extra duwtje had gegeven.

We kunnen het experiment van Lissajous modelleren door in het x,y-vlak een punt zowel een horizontale harmonische beweging $x = a\cos(\alpha t)$ als een verticale beweging $y = b\sin(\beta t)$ te laten uitvoeren. Er is in principe niet veel meer voor nodig dan het uiteinde van een slinger te voorzien van een schrijfstift, maar het is toch een moeizaam werkje. Op school hebben sommigen wellicht een soortgelijk experiment in het natuurkundelokaal zien uitvoeren met behulp van een kathodestraaloscillograaf.

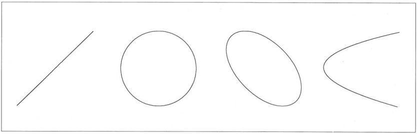
Maar nu, met de computer in de huiskamer, kan het experiment op uiterst eenvoudige wijze uitgevoerd en gevarieerd worden. Eigenlijk doen we hetzelfde als wat vroeger de natuurkundedocent deed, want de plaatjes op het beeldscherm zijn eveneens het resultaat van heen en weer schietende elektronenstralen.

Op het beeldscherm kunnen we a en b als eenheid in horizontale respectievelijk verticale richting nemen. De te gebruiken formules zijn dan

$$x = \cos(\alpha t), \qquad y = \sin(\beta t)$$

Voorlopig nemen we voor α en β positieve gehele getallen. De eenvoudigste keuze is $\alpha = 1$ en $\beta = 1$. Zonder de computer te gebruiken, met alleen enige kennis van de eigenschappen van goniometrische functies, kunnen we voorspellen dat het resultaat een cirkel is. Er is immers voldaan aan $x^2 + y^2 = 1$. Zouden we een andere keuze van lengte-eenheden hebben gemaakt, dan zouden we een ellips hebben gekregen.

Laten we nu eens $\alpha = 2$ en $\beta = 1$ nemen. De goniometrie leert ons dat in dat geval voldaan is aan $x = 1 - 2y^2$. Wie op school opgelet heeft herkent hier de beschrijving van een parabool. Op school waren de x en de y verwisseld en was er sprake van een verticale parabool, een parabool met een verticale symmetrieas. Hier wordt het een horizontale parabool. Maar dat is nog niet alles. Een parabool strekt zich als meetkundige figuur oneindig ver uit en dat kan hier beslist niet. De natuurkundige ziet dat aan de eindige slingerlengte bij het experiment van Lissajous en de wiskundige



figuur 2.1 Een paar Lissajous-figuren

merkt op dat zowel de sinusfunctie als de cosinusfunctie slechts waarden tussen -1 en +1 kunnen aannemen. Hij weet dus dat een willekeurige keuze van α en β altijd aanleiding geeft tot een kromme die niet buiten het eenheidsvierkant kan treden. Altijd geldt dus

$$-1 \leq x \leq 1 \qquad \text{en} \qquad -1 \leq y \leq 1$$

In het genoemde geval bestaat de Lissajous-kromme kennelijk uit een boog van de parabool welke tweemaal doorlopen wordt.

Het speciale geval $\alpha = 2$ en $\beta = 2$ verschilt nauwelijks van dat van $\alpha = 1$ en $\beta = 1$. In beide gevallen is het een cirkel met straal 1, maar alleen de tijdschaal is anders. In het tweede geval is de slingering tweemaal zo snel. Letten we alleen op de meetkundige vorm van de Lissajous-kromme, dan is alleen de onderlinge verhouding van α en β van belang. Anders gezegd, we kunnen zonder verlies van algemeenheid $\beta = 1$ kiezen en α als een breuk van positieve gehele getallen opvatten, zeg $\alpha = m/n$. Zijn m en n zo klein mogelijk gekozen, dat wil zeggen kan de breuk m/n niet meer vereenvoudigd worden, dan kunnen we constateren dat x en y na verloop van $2n\pi$ eenheden van tijd weer dezelfde waarden aannemen. Dat betekent dat de Lissajous-kromme een gesloten figuur is. In dat tijdsverloop zijn er horizontaal m complete slingeringen geweest tegenover n verticaal. In de figuur kunnen we die getallen terugvinden door te letten op de uiterste waarden van x en van y, meetkundig dus de raakpunten van de kromme aan de zijden van het omsloten vierkant.

Tot dusverre hebben we horizontaal een cosinusfunctie met verticaal een sinusfunctie gecombineerd. In feite betekent dat een faseverschil van 1/4 tussen beide trillingsvormen. Immers, voor $t = 0$ is de horizontale uitwijking maximaal, terwijl de verticale slingering door zijn nulpunt gaat. Dat kan met weinig moeite algemener, en aldus komen we tot het volgende computermodel:

$$x = \cos(st), \qquad y = \cos(t + 2f\pi)$$

Letten we alleen op de vorm van de Lissajous-kromme en nemen we een eventuele spiegeling of kwartdraai voor lief, dan kunnen we $s \geq 1$ nemen

en f, de fase, beperken tot het interval [0,1/2].

De eenvoudigste situaties, de reeds besprokene en een paar andere, vatten we even samen (zie figuur 2.1):

$s = 1, f = 0$	lijnstuk
$s = 1, f = 1/4$	cirkel
$s = 1, 0 < f < 1/2$	ellips
$s = 2, f = 0$ of $f = 1/2$	parabool

Met gebruikmaking van het programma LISSA1 kunnen we dan reeds een groot aantal eenvoudige experimenten uitvoeren.

De gulden snede

Bij de gegeven versie van het programma moet men van tevoren zelf de te gebruiken parameterwaarden in het programma invoeren. Wil men het plaatje aanpassen aan de rechthoekige afbeelding van het beeldscherm, dan kan men $a = 4/3$ nemen. In het programma gaat de slinger net zolang door met punten op het beeldscherm neer te zetten, totdat men er genoeg van krijgt. Met een willekeurige toetsaanslag, de spatiebalk bijvoorbeeld, wordt het programma beëindigd. Men kan dan het programma herhalen met andere parameterwaarden of het beëindigen.

Heeft men voor s een keuze gemaakt van een breuk m/n met kleine getallen m en n, dan wordt de Lissajous-kromme na enige omlopen gesloten, zoals we hierboven reeds hebben opgemerkt. Maar het is aardig ook eens te experimenteren met een 'onmeetbare' frequentieverhouding. Weliswaar is elk getal in een computer meetbaar en altijd voorstelbaar als een breuk van gehele getallen, maar bij een getal als $s = (-1 + \sqrt{5})/2$, het bekende *gulden-snedegetal*, zijn teller en noemer van de daarbij behorende breuk dusdanig groot dat het toch net is alsof s onmeetbaar is.

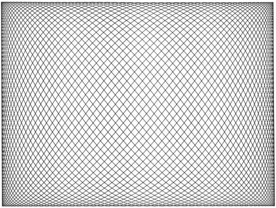
Hoe dan ook, men moet maar eens het speciale geval met die waarde en met $a = 1$ (of $a = 4/3$) en $f = 0$ uitvoeren om te zien wat er gebeurt. Het resultaat is een Lissajous-kromme die zich nimmer sluit en die een vierkant of rechthoek gelijkmatig vult.

Mooie plaatjes krijgt men wanneer m en n teller en noemer zijn van benaderende breuken van het gulden-snedegetal. Ze kunnen worden afgelezen uit de bekende rij van Fibonacci, een heel oud model van ongeremde groei:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$$

Een voorbeeld zien we in figuur 2.2, waarin $m = 34$ en $n = 55$ gekozen is. Met nog hogere getallen uit de rij van Fibonacci verandert de figuur kwalitatief niet veel meer en neemt alleen de mate waarin het vlak regelmatig wordt gevuld nog meer toe. Men kan het zelf aanschouwen op het beeldscherm met behulp van LISSA2, het programma waarmee figuur 2.2 gemaakt is.

figuur 2.2
Een Lissajous-figuur met $m = 34$ en $n = 55$



Generalisaties

De figuren van Lissajous zijn het eenvoudigste voorbeeld van een familie van krommen die beschreven kunnen worden door

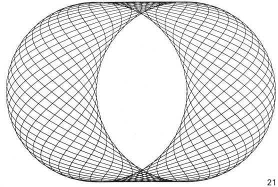
$$x = f(t), \qquad y = g(t)$$

waarbij zowel f als g de som is van een paar sinus- of cosinustermen. Eenvoudige gevallen geven op het beeldscherm al verrassende resultaten. In figuur 2.3 tonen we het resultaat van

$$x = a\cos(t) + b\cos(st), \quad y = \sin(t)$$

voor het geval $a = 1$, $b = 0.5$, $s = 2.45$. Het bijbehorende programma LISSA3 is een variant van het programma LISSA1.

figuur 2.3
Een van de vele verwanten van de Lissajous-familie



Bloemvormen

Lissajous-achtige krommen kunnen heel goed worden gebruikt om bepaalde bloemvormen te simuleren, in het bijzonder wanneer de bloemkroon een rotatiesymmetrie bezit. In dat geval moeten we uitgaan van een beschrijving in poolcoördinaten r en φ, waarbij r en φ periodieke functies zijn van een variabele t die desgewenst als de tijd geïnterpreteerd kan worden. We geven slechts een eenvoudig voorbeeld en we laten het aan de lezer over hierop voort te borduren. Willen we bijvoorbeeld een bloem met een zestallige draaiingssymmetrie imiteren, dan moet telkens wanneer φ met 2π toegenomen is r na zes golvingen weer op zijn uitgangswaarde zijn teruggekeerd. De eenvoudigste formules zijn in de trant van

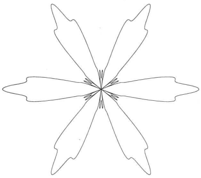
$$r = a + b\cos(6t), \qquad \varphi = t$$

waarbij a en b positieve getallen zijn.

Omdat de voerstraal normaal gesproken een positieve waarde dient te hebben moeten we $a > b$ kiezen, dus bijvoorbeeld $a = 1$ en $b = 0.8$. We weten dan zonder tekenen al dat de insnijdingen van de bloembladen tot 1/9 van hun lengte kunnen reiken. De maximale waarde van r is immers a+b, terwijl a-b de minimale waarde is. In het programma LISSAF doen we het een klein beetje algemener door aan r nog een tweede goniometrische term, een hogere harmonische, toe te voegen. In het programma moeten bij het tekenwerk de poolcoördinaten in Cartesische coördinaten omgerekend worden met behulp van de formules

$$x = r\cos\varphi, \qquad y = r\sin\varphi$$

In figuur 2.4 laten we een van de vele resultaten zien die met dit programma verkregen kunnen worden. In dat programma kunnen de parameterwaarden a_1 en a_2 worden veranderd. In plaats van 6 en 36 voor de parameters s_1 en s_2 kan men bijvoorbeeld 5 en 25, een vijftallige symmetrie, enzovoort nemen. De speciale keuze van de parameters a_1 en a_2 geeft in dit voorbeeld aanleiding tot negatieve waarden van r. Voor de meetkundige afbeelding maakt dat technisch gesproken niet zoveel uit, maar het resultaat is wel iets dat doet denken aan meeldraden!

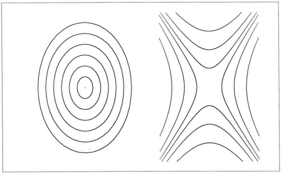


figuur 2.4 Simulatie van een bloemblad

3 Wandelkaart en borduurpatroon

Wie bij een wandeling in een berglandschap een goede wandelkaart gebruikt heeft veel plezier van de daarop aangegeven hoogtelijnen, niveaulijnen of isohypsen. Langs een dergelijke lijn is gemeten ten opzichte van het zeeniveau de hoogte overal gelijk. De omgeving van een heuvel wordt bijvoorbeeld gemarkeerd door een aantal ellipsvormige niveaulijnen, waarvan de hoogte telkens een vaste afstand, zeg tien meter, verschilt (zie figuur 3.1). Heeft de top van de heuvel een vlakke vorm terwijl de hellingen vrij steil zijn, dan kunnen we dat aan de kaart zien. De niveaulijnen liggen dicht tegen elkaar aan waar de helling steil is, en ze liggen ver uit elkaar wanneer we het vlakke topgedeelte naderen. Bevinden we ons op een bergpas, dan zouden we heel wat anders zien. Lopen we langs een pasweg en zijn we op het hoogste punt aangeland, dan zien we links en rechts omhooggaande hellingen terwijl het voor en achter ons omlaag gaat. Op de kaart zien we dan een aantal hyperboolachtige bogen als geïllustreerd in figuur 3.1.

Het is vrij eenvoudig hiervan met de computer een simulatie te maken, maar daarvoor is wel enige elementaire kennis van de analytische meetkunde van de ruimte nodig. De hoofdzaak is dat we aan een functie $f(x,y)$ van twee variabelen een meetkundige betekenis kunnen verbinden. Het helpt misschien wanneer we de variabelen x en y zien als beeldschermcoördinaten van een Cartesisch coördinatenstelsel terwijl de z-as er loodrecht op staat, naar ons toegericht. In de analytische meetkunde van de ruimte beschikken we over een paar standaardoppervlakken als bol, kegel, paraboloïde, ellipsoïde en zadelvlak. Een bol met straal 1 is bepaald door de vergelijking



figuur 3.1
Berg en pas

$$x^2 + y^2 + z^2 = 1$$

een drieassige ellipsoïde door

$$(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$$

een (elliptische) paraboloïde door

$$z = (x/a)^2 + (y/b)^2$$

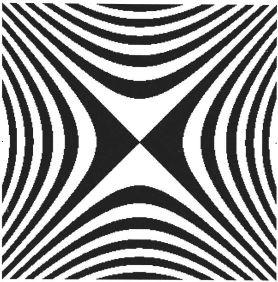
en een zadelvlak (hyperbolische paraboloïde) door

$$z = (x/a)^2 - (y/b)^2$$

Voor de simulatie van een heuvel kunnen we een paraboloïde nemen, zeg $z = 1 - x^2/4 - y^2$. We stellen ons voor dat $z = 0$ het niveau van de laagvlakte is en dat de paraboloïde als een soort reflector van een fietslamp erboven uitsteekt. Om de bijbehorende hoogtekaart te maken gebruiken we het programma HOOGTE1.

In dat programma denken we het oppervlak uitgespreid over het vierkant $-1 < x < 1, -1 < y < 1$ als grondvlak. Dat vierkant verdelen we in een flink aantal punten, $(2n + 1)^2$ voor $n = 160$ bijvoorbeeld, dus ongeveer 100 000. In elk van die punten bepalen we de hoogte z door $z = x^2 - y^2$ uit te rekenen. Het oppervlak verdelen we in horizontale plakjes, een aantal

figuur 3.2 Hoogtelijnen in de omgeving van een bergpas



van m over een hoogteverschil van 1. Die plakjes kunnen we nummeren met een geheel getal door uit z het gehele getal int(mz) af te leiden. In het grondvlak, het computerscherm dus, markeren we alleen de punten die tot plakjes met een even rangnummer behoren. Aldus krijgen we een zwart-wit patroon waarvan de randen, in dit speciale geval bogen van orthogonale hyperbolen, de hoogtelijnen zijn. Het resultaat, dat we kunnen interpreteren als de afbeelding van de omgeving van een bergpas, is afgebeeld in figuur 3.2.

Het programma kan op een aantal manieren gevarieerd worden. Uiteraard kunnen we de parameters n en m veranderen. Interessanter is het om met andere functies te experimenteren, maar daarbij moeten we er wel op letten dat z voor alle waarden van x en y binnen het vierkant betekenis heeft. Een functievoorschrift als $z = \sqrt{(1 - x^2 - y^2)}$ is onvoldoende omdat een wortel uit een negatief getal geen betekenis heeft. Om dat probleem te ondervangen moet de definitie worden uitgebreid tot bijvoorbeeld

$$\text{if} \quad x^2 + y^2 < 1 \quad \text{then} \quad z = \sqrt{(1 - x^2 - y^2)} \quad \text{else} \quad z = 0$$

Met deze keuze krijgen we de hoogtekaart van een halve bol. De hoogtelijnen zijn dan natuurlijk cirkels. Dat zelfde geldt ook voor de keuze $z = x^2 + y^2$, model van een dal. Men bestudere zelf waarin deze hoogtekaart verschilt van de vorige. Ten slotte kan men dan experimenteren met $z = \sqrt{(x^2 + y^2)}$, een kegel met de punt naar beneden.

Wie over kleuren kan beschikken kan de verschillende laagjes elk een eigen kleur geven. Normaal zijn er in de veelgebruikte EGA/VGA-modus zestien kleuren aanwezig, waaronder wit en zwart. Men kan daaruit een zeker aantal kiezen, zeg tien. Zijn er meer dan tien laagjes, dan kunnen die kleuren telkens worden herhaald. In het programma HOOGTE2 wordt het een en ander gerealiseerd (zie figuur 3.3, bladzijde 81 in het kleurkatern).

De vermelde programma's staan model voor een groep programma's die in grote lijnen met elkaar gemeen hebben dat in een rechthoek van pixels voor elke individuele pixel de kleurwaarde wordt bepaald volgens een zelfde voorschrift. Op soortgelijke wijze worden ook de fraaie details van de Mandelbrot-verzameling verkregen. Andere versies van dit programma zullen we later tegenkomen.

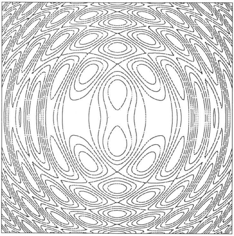
Een professionele hoogtekaart

Het is geen eenvoudig wiskundig probleem om de niveaulijnen te bepalen van een willekeurig wiskundig oppervlak $z = f(x,y)$. In de voorafgaande programma's hebben we de niveaulijnen eigenlijk op indirecte wijze bepaald als randen van gebieden waar de hoogte z tussen twee nauwe grenzen lag. Met brute kracht hebben we de hoogte bepaald in een groot aantal punten in de orde van een paar honderdduizend. Alleen met gebruikmaking van de moderne huiscomputers kan men binnen redelijke tijd op resultaat rekenen. Veel beter is er eigenlijk niet, of er zou nog meer gerekend moeten worden. Een van de beste en meest eenvoudige programma's – LEVEL – beschrijven we hieronder.

In dat programma nemen we aan dat het oppervlak $z = f(x,y)$ zich uitstrekt over een gegeven rechthoek in het x,y-vlak en dat we het gedeelte $-1 < z < 1$ in een nader te bepalen aantal niveaus willen verdelen. Het komt erop neer dat we het oppervlak benaderen door een netwerk van kleine driehoekjes. Elk driehoekje is een plat vlak. Wiskundig is het niet moeilijk om van een plat vlak, dat in het algemeen een schuine stand heeft, niveaulijnen te bepalen. Daartoe moet steeds hetzelfde meetkundige probleempje worden opgelost, de snijlijn te bepalen van een driehoekig vlakdeel en een horizontaal plat vlak. Eerst verdelen we het rechthoekige grondvlak in een rooster van kleine rechthoekjes. Elk rechthoekje is het grondvlakje van een staand blokje, een dunne kolom met een rechthoekige doorsnede, die het oppervlak doorsnijdt. Daarbij berekenen we alleen de hoogte van de vier snijpunten S_1, S_2, S_3 en S_4 op de hoekpunten. Die snijpunten hoeven niet noodzakelijk in een zelfde vlak te liggen wegens de in het algemeen gebogen vorm van het oppervlak. Aan die vier punten voegen we het zwaartepunt Z toe, dat correspondeert met de gemiddelde hoogte in de hartlijn van de toren. Voor elk kolommetje beschikken we nu over een viertal driehoekjes ZS_1S_2, ZS_2S_3, ZS_3S_4 en ZS_4S_1 die met elkaar een triangulatie van een stukje oppervlak vormen. Het probleem om een driehoekje met een willekeurig horizontaal vlak te snijden is in principe eenvoudig, maar er zijn daarbij verschillende situaties mogelijk die elk een eigen berekeningswijze vereisen. In het programma zijn die mogelijkheden via een 'multiswitch' ondergebracht in een 'select case'-constructie.

In figuur 3.4 is aldus een hoogtekaart gemaakt van een oppervlak bepaald door de volgende formule:

$$f(x,y) = \sin(x^2 + y^2 - 2x + 1)\,\sin(x^2 + y^2 + 2x + 1)$$



figuur 3.4
Niveaulijnen van een oppervlak

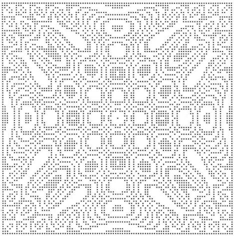
Borduurpatronen

Men zou kunnen denken dat handwerken een typisch vrouwelijke bezigheid is terwijl het werken met een computer meer iets voor een man is. En misschien is het in sommige huishoudens zo dat vrouwen wanhopig worden omdat hun echtgenoot of partner niet van de computer weg te branden is. Wellicht dat dit hoofdstuk ertoe kan bijdragen de wederzijdse band tussen man en vrouw te verstevigen. Met de computer is men namelijk in staat een onuitputtelijk aantal fraaie borduurpatronen te ontwerpen die uitnodigen tot het maken van bijvoorbeeld tafelkleedjes. We gaan uit van het programma PATROON1 dat veel lijkt op HOOGTE1.

In het nieuwe programma kan het niveaugetal c via het toetsenbord worden ingevoerd. Het is de bedoeling dat men het programma eerst met een paar lage waarden van c uitvoert, $c = 5$ of $c = 10$ bijvoorbeeld, en dat men daarna met een paar flinke tussenstappen naar waarden van zeg $c = 500$ toegaat. De af te beelden figuur is in dit programma wat ruimer doordat we bij het plotten telkens een paar pixelposities overslaan. De gekozen functie $z = (1 - x^2)(1 - y^2)$ hoort niet bij een oppervlak van een bekend type, maar we kunnen er zonder tekenen wel een paar bijzonderheden over opmerken. Aan de rand van het vierkant $-1 < x < 1, -1 < y < 1$ is $z = 0$, maar binnen het vierkant is z positief met het maximum $z = 1$ in de oorsprong. Verder is het oppervlak symmetrisch als de letter X: spiegelsymmetrisch ten opzichte van de x-as en de y-as en ten opzichte van de diagonalen.

Kiezen we $c = 10$ dan zien we op het scherm weer een hoogtekaart van het oppervlak ontstaan, maar wanneer c erg groot is lijkt de figuur niets meer met een hoogtekaart van doen te hebben. In figuur 3.5 hebben we het patroon afgebeeld dat voor $c = 256$ ontstaat.

figuur 3.5
Van hoogtekaart tot borduurpatroon



Zonder twijfel een fraai patroon, maar als hoogtekaart voor een bergwandeling volslagen onbruikbaar. Het is niet moeilijk om de oorzaak daarvan te achterhalen. Laten we gemakshalve $c = 1000$ nemen. Dat betekent dat met de factor c de hoogte cz zich nu uitstrekt van de waarde 0 tot 1000. In de x-richting is er maar plaats voor zo'n honderd punten en die kunnen onmogelijk de duizend verschillende niveaus weergeven. Zouden we voor een bepaalde waarde van x en y een hoogte vinden van $cz = 576.38...$ of afgerond 576, dan moet er op de overeenkomstige plaats als aangegeven door de tellers i en j kennelijk een punt worden afgebeeld omdat dat getal even is. Gaan we een stapje naar links of naar rechts, dat wil zeggen kijken we naar de posities i-1 of i+1 met gelijke j, en rekenen we wederom cz uit, dan kan de uitkomst behoorlijk afwijken, we kunnen er best een tiental niveaus naast zitten; het zou 583.62... of 584.38... of zoiets kunnen zijn. Het is in het algemeen onvoorspelbaar of het gehele deel van de uitkomst even of oneven is. Kortom, in het patroon dat we krijgen zijn de posities waar int(cz) even is min of meer onvoorspelbaar. Dat betekent chaos, en die chaos neemt alleen maar toe wanneer c groter wordt.

We kunnen dit betoog aldus samenvatten:

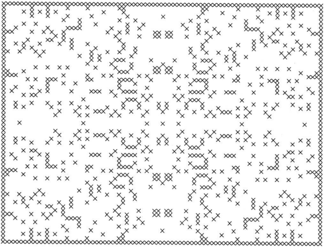
Bij lage waarden van c *behoort orde, behoort een hoogtekaart.*
Bij hoge waarden van c *behoort chaos, behoort een borduurpatroon.*

Het programma kan nog wat verbeterd worden door echt kruisjes te plaatsen waar eerst punten waren afgebeeld. We krijgen dan het programma PATROON2 (zie figuur 3.6). Er zijn daarin nog een paar extra parameters aangebracht met behulp waarvan het patroon kan worden beïnvloed. De printvoorwaarde hangt af van de gehele waarde van cz. Tot dusverre werd er alleen iets afgebeeld wanneer int(cz) even was. Dat kan natuurlijk ook anders, bijvoorbeeld alleen iets afdrukken wanneer int(cz) oneven is of wanneer int(cz) een drievoud is.

In het programma is dit algemeen een m-voud en is hier $m = 5$ gekozen. Gemiddeld wordt er van de vijf mogelijke posities slechts een afgebeeld. Het resulterende patroon is dus een beetje ijl, maar dat kan juist een bedoeld effect zijn. De grootte van de kruisjes hangt af van de parameter h. Met de parameter d worden de afstanden tussen de kruisjes en tevens de grootte van het hele patroon verkleind of vergroot. Voor d kan men een klein geheel getal nemen, tussen 1 en 6 bijvoorbeeld. Met de parameters n_1 en n_2 kunnen we de grootte van het patroon nader regelen. Men kan het programma toepassen voor een veelheid van functies $z = f(x,y)$.

Omdat dit programma symmetrisch is ten opzichte van x en y is het voldoende een expliciet functievoorschrift te geven voor waarden van x en van y in het interval (0,1). Geschikte mogelijkheden zijn

$$z = x^2 + 2y^2$$
$$z = 1 - xy$$
$$z = xy(1 - x)(1 - y)$$
$$z = x^4 - 6x^2y^2 + y^4$$
$$z = \cos(4\pi x)\cos(3\pi y)$$



figuur 3.6 Een kruissteekjespatroon

Bouwen met bergen

Men kan natuurlijk op goed geluk allerlei functies proberen, maar men kan ook de omgekeerde weg inslaan door aan te sturen op een berglandschap dat een bepaalde meetkundige vorm dient te hebben. Men kan daartoe het landschap als het ware opbouwen uit bergen als elementaire bouwstenen. Een berg opgericht op het x,y-vlak met de top in de oorsprong kan worden voorgesteld door de functie

$$z = \exp(-x^2 - y^2)$$

waarbij exp de notatie is van de exponentiële functie.

Men ziet zonder veel moeite in dat de berg cirkelsymmetrisch is, dat z tussen 0 en 1 inligt met $z = 1$ voor de top in de oorsprong. De hoogtelijnen zijn cirkels. Om te weten hoever de berg zich uitstrekt kan men gebruik maken van de gegevens

$$\exp(-2.3) \approx 1/10, \qquad \exp(-4.6) \approx 1/100$$

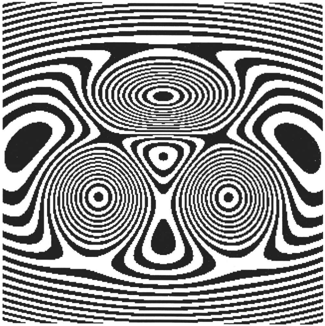
Dat betekent dat op een afstand van ongeveer 1.5 van de top de hoogte nog maar 1/10 is van die van de top. De berg kan men nog een beetje modelleren door x^2+y^2 te vervangen door een algemenere kwadratische vorm, dus

$$z = \exp(- ax^2 - bxy - cy^2)$$

waarbij a en c positief zijn. Een geschikte keuze van de drie constanten a, b en c geeft dan een berg met een elliptische doorsnede. Is $b = 0$ dan is de ellipsvorm horizontaal of verticaal gericht, is $a = b$ dan is de ellipsvorm diagonaal gericht.

Bij wijze van voorbeeld stellen we in het programma PATROON3 een landschap samen uit drie bergen geplaatst op een landschap dat enigszins bolvormig is. In figuur 3.7 laten we zien wat het resultaat is. Uiteraard kan men in dit voorbeeld ook nog veel wijzigingen aanbrengen. De hoofdzaak is dat er voldoende materiaal beschikbaar is om mee te experimenteren, mee te simuleren.

We besluiten dit hoofdstuk met een kleurenversie – PATROON4 – waarin een rechthoek op quasi-toevallige wijze wordt gevuld met kleurige pixels bepaald door de functie $c(1 - x^2)(1 - y^2)$ waarin c een groot getal is. Het in figuur 3.8 (zie bladzijde 81) weergegeven resultaat vertoont een opmerkelijke tekening die als *moiré* bekend staat. In het volgende hoofdstuk gaan we uitvoerig op dat verschijnsel in.



figuur 3.7 Een hoogtekaart van een berglandschap

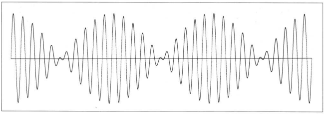
4 Zwevingen en moiré

De lezer is ongetwijfeld vertrouwd met het optreden van zwevingen. Een musicus kent het hinderlijke verschijnsel dat de combinatie van twee tonen voortgebracht door trillingsbronnen waarvan de frequenties heel weinig verschillen aanleiding geeft tot een periodiek aanzwellen en verzwakken van het geluid. Een pianostemmer maakt er handig gebruik van door bij het stemmen juist op dergelijke zwevingen te letten. Heeft de ene snaar bijvoorbeeld een frequentie van 500 trillingen per seconde en een naastliggende snaar er een van 502 Hertz dan constateert een goede luisteraar, en een pianostemmer is zo iemand, tweemaal per seconde een versterking en een verzwakking. Het is dus de kunst bij het stemmen zodanig te werk te gaan dat het zwevingseffect geheel verdwijnt. Het verschijnsel kan eenvoudig worden verklaard door twee hoogfrequente harmonische trillingen bepaald door de formules $\sin(\alpha t)$ en $\sin(\beta)$ bij elkaar op te tellen. De goniometrie leert ons dat

$$\sin(\alpha t) + \sin(\beta t) = 2\sin((\alpha + \beta)t/2)\cos((\alpha - \beta)t/2)$$

Verschillen α en β heel weinig dan kunnen we het rechterlid opvatten als in hoofdzaak een harmonische trilling waarvan de frequentie het gemiddelde is van α en β, dat dus weinig van α en β verschilt. De amplitude $2\cos((\alpha\text{-}\beta)t/2)$ is daarentegen een langzaam veranderende, laagfrequente, harmonische trilling die $\alpha - \beta$ maxima per seconde oplevert. Wie formules graag bevestigd ziet door plaatjes op het computerscherm kan zich overtuigen met een computerexperiment. Daarbij kan men het programma ZWEVING gebruiken waarin de som van twee harmonische trillingen $\sin(2\pi s_1 t)$ en $\sin(2\pi s_2 t)$ grafisch wordt afgebeeld. Het resultaat ziet men in figuur 4.1.

Essentieel voor het zwevingseffect is het optreden van *verschiltonen*. Een soortgelijk verschijnsel kan men waarnemen tijdens televisiepro-



figuur 4.1 Zwevingen bij naburige tonen

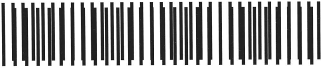
gramma's wanneer het beeldscherm niet nauwkeurig is afgesteld. Men ziet dan streperige banden over het scherm lopen, tenminste voor bepaalde stations met hoge frequenties. Op het scherm ziet men dan de minder prettige resultaten van laagfrequente verschilfrequenties. Er kunnen voor dat resonantieverschijnsel verschillende oorzaken zijn: een oud toestel, een onjuiste bekabeling of fouten bij het doorgeven van het signaal, om maar iets te noemen; hinderlijk blijft het in elk geval.

Hekken en gordijnen

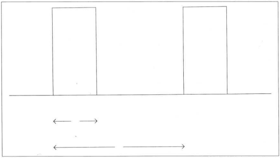
Minnaert, een Utrechtse hoogleraar, astronoom en fysicus, die veel heeft gedaan aan de popularisering van de wetenschap, heeft in zijn beroemde trilogie *De natuurkunde van het vrije veld* het optreden van zwevingen tussen twee hekken beschreven. In zijn woorden: 'Overal waar men twee hekken, het ene door het andere heen ziet, bemerkt men brede lichte en donkere golvingen in de lichtsterkte, die verlopen zodra men beweegt.' In figuur 4.2 zien we een illustratie van dat verschijnsel. Met behulp van het computerprogramma HEK kunnen we zelf wat experimenten uitvoeren.

In het programma is verondersteld dat we kunnen werken in een modus waarbij op het beeldscherm 640 verticale pixelrijen beschikbaar zijn. Verstaan we onder weglengte a de som van latbreedte en grootte van opening (zie figuur 4.3), dan kan a bijvoorbeeld tussen 8 en 32 worden gekozen met een latbreedte b van $a/2$ of kleiner. In figuur 4.2 is $a_1 = 19$ en $a_2 = 22$ gekozen met een latbreedte van 5. Op het scherm kunnen we dan hooguit een dertigtal spijlen zien, wat ruim voldoende is voor het optreden van het zwevingseffect.

Zoals Minnaert reeds opmerkte treedt dat zelfde verschijnsel ook op wanneer we twee lagen fijngeweven textiel op elkaar leggen. Bij in plooien

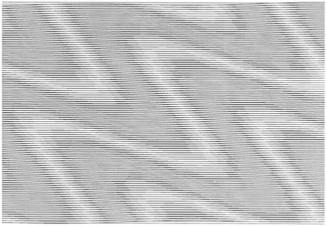


**figuur 4.2
Zwevingen
bij twee hekken**



**figuur 4.3
Repeterend
element van hek
of moiré-patroon**

figuur 4.4
Moiré bij evenwijdige lijnen

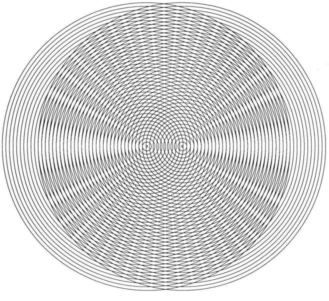


hangende vitrage kunnen we dat effect heel goed waarnemen, maar ook wordt er om esthetische redenen in de textielindustrie bewust gebruik van gemaakt. Men legt twee weefsels in vochtige toestand op elkaar en men voert ze dan onder grote druk tussen twee hete cilinders door zodanig dat de draden plaatselijk geplet worden. Lichtreflectie geeft dan het merkwaardig golvende effect dat men moiré noemt. De fraaiste tekening krijgt men bij weefsels van zijde of rayon. In de beeldende kunsten is moiré een van de methoden van vooral op-art kunstenaars om een soort beweging te suggereren. Figuur 3.7, een soort hoogtekaart van een berglandschap, is ook een voorbeeld van dat verschijnsel. Daar zit overigens heel wat achter, maar daar komen we aanstonds uitvoerig op terug.

In figuur 4.4 zijn we uitgegaan van een stelsel evenwijdige lijnen die onderling een gelijke afstand hebben, equidistant zeggen we. Daarop is een rooster geplaatst dat maar heel weinig van het vorige verschilt. De lijnen zijn niet recht meer maar heel licht golvend. Interferentie van de twee roosters leidt in dit geval tot een eigenaardig moiré-patroon van zigzaglijnen. Dergelijke patronen kan men heel eenvoudig met een laserprinter zelf te voorschijn roepen door een zelfde rooster tweemaal af te drukken, het eerste op papier en het tweede op een transparant zoals dat gebruikt wordt voor een overhead-projector. Legt men het transparant bovenop de papierafdruk zodanig dat beide patronen net niet tot dekking komen, dan ziet men fraaie moiré-patronen ontstaan. In de programma's MOIREL1 en MOIREL2, waarin het een en ander gesimuleerd wordt, ziet men overeenkomstige patronen ontstaan.

Cirkelpatronen

In figuur 4.5 zijn we uitgegaan van een bundel concentrische cirkels waarbij de stralen lineair toenemen, dus ook weer equidistant. Die bun-



figuur 4.5 Moiré-patroon bij twee cirkelschijven

del laten we interfereren met een kopie van dezelfde cirkelschijf, maar dan iets verschoven. Het moiré-patroon bestaat nu uit een aantal stralen door het centrum van de figuur. De figuur kan men op het computerscherm nabootsen met behulp van het programma MOIREC1.

Wie dit programma met de computer uitvoert zal meestal getuige zijn van het optreden van een extra moiré-effect doordat de tekening, in dit geval een stelsel concentrische cirkels, gaat interfereren met de pixelposities van het beeldscherm. We komen daar aanstonds op terug, maar voorlopig doen we even alsof het beeldscherm een heel hoog oplosbaar vermogen heeft. Wanneer we de stralen van de cirkels niet lineair laten toenemen maar bijvoorbeeld volgens een wortelformule als in het programma MOIREC2, krijgen we een ander moiré-patroon te zien, verticale evenwijdige lijnen. De vraag is hoe we dat kunnen begrijpen en eventueel of we een moiré-patroon ook kunnen voorspellen.

Een beetje theorie

Om het moiré-effect echt te kunnen begrijpen dient de lezer over iets meer kennis van wiskunde te beschikken. Voor een deel is dat dezelfde kennis die men in het vorige hoofdstuk nodig had om een hoogtekaart met niveaulijnen van een wiskundig oppervlak te kunnen begrijpen. Dat is dus een beetje coördinatenmeetkunde in drie dimensies. Wanneer men ook nog wat verstand heeft van de beginselen van de differentiaalrekening

zal men tot het verrassende inzicht kunnen komen dat het vormen van een moiré-patroon een soort grafische interpretatie van differentiëren is. Overigens kan men zonder bezwaar de eventueel moeilijke passages overslaan en zich meteen met de computersimulaties bezighouden.

De sleutel tot de theorie is de interpretatie van lijnenroosters als de niveaulijnen van een oppervlak, een hoogtekaart van een berglandschap zoals beschreven in het vorige hoofdstuk. Om niet in herhaling te vervallen stippen we de belangrijkste elementen hier even aan. Meer bijzonderheden kan men in het vorige hoofdstuk vinden. Op een vlakke helling lopen de niveaulijnen nagenoeg evenwijdig. In de omgeving van een heuvel zien de hoogtelijnen eruit als concentrische ellipsen, en bij passen zijn het hyperbolische bogen. Is $z = f(x,y)$ de vergelijking van het oppervlak, dan zijn de niveaulijnen bepaald door $f(x,y) = nh$ waarbij n de rij van de gehele getallen doorloopt en h meestal een klein getal is. Bij elke functie, en een keuze van h, behoort een lijnenrooster. Het eenvoudige geval

$$f(x,y) = ax$$

waarbij a een willekeurig getal is, geeft door gelijkstelling van ax met nh een stelsel equidistante evenwijdige lijnen bepaald door

$$x = nh/a$$

het meest eenvoudige lijnenrooster. De constante a doet er kennelijk niet zoveel toe, alleen de onderlinge verhouding van a en h is van belang. De keuze

$$f(x,y) = ax + by$$

leidt analoog tot een equidistant lijnenrooster in een scheve positie. De kegelkeuze

$$f(x,y) = \sqrt{(x^2 + y^2)}$$

leidt tot

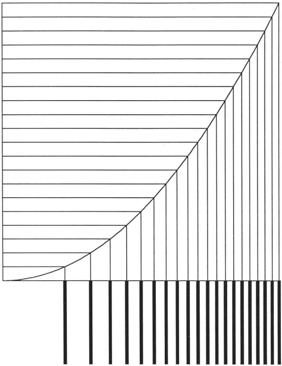
$$\sqrt{(x^2 + y^2)} = nh$$

een stelsel equidistante concentrische cirkels.

Het principe hoe roosterlijnen kunnen worden opgevat als niveaulijnen hebben we meetkundig geïllustreerd in figuur 4.6 voor een eendimensionaal rooster waarbij de roosterafstand volgens een wortelformule verloopt. De positie van de roosterlijnen is dan bepaald door een formule van de vorm

$$x = a\sqrt{n}$$

Omdat bij de hoogtelijnen de hoogte lineair, dat wil zeggen evenredig met n, toeneemt behoeven we uit die formule slechts n uit te drukken in x. Dat geeft $n = x^2/a^2$. Die formule moet equivalent zijn met $f(x,y) = nh$.



figuur 4.6
Roosterlijnen als niveaulijnen

Het resultaat is dus

$$f(x,y) = (h/a^2)x^2$$

of gewoon x^2 wanneer we de schalingsfactor negeren. In de figuur zien we hoe de parabool $x \rightarrow x^2$ als de doorsnede van een dal door equidistante horizontale niveaulijnen doorsneden wordt en hoe dat aanleiding geeft tot een verticaal lijnenrooster.

Het moiré-theorema

We beschouwen nog eens het eenvoudigste geval van een equidistant lijnenrooster waarbij $f(x) = ax$. Voor een toon zou dat corresponderen met een frequentie a en een golflengte h/a. We laten dat patroon interfereren met een iets verschillend equidistant rooster $g(x) = bx$ waarvan de frequentie weinig van a verschilt. Het aantal optredende zwevingen is bepaald door de verschiltoon waarvan de frequentie $|a\text{-}b|$ is. In de meetkun-

dige terminologie wil dat zeggen dat het moiré-patroon dat door de interferentie van $f(x)$ en $g(x)$ ontstaat gelijk is aan dat van het verschil $|f(x) - g(x)|$. Het ligt voor de hand dat die zelfde uitspraak ook waar is wanneer $f(x)$ en $g(x)$ niet noodzakelijk lineair zijn. Voegen we de tweede variabele y er ook nog aan toe dan krijgen we de volgende fundamentele eigenschap:

> *Het moiré-patroon van* f(x,y) *en* g(x,y)
> *is bepaald door* |f(x,y) - g(x,y)|.

We zijn nu in staat te begrijpen waarom we in het voorbeeld van MOIREC2 als moiré-patroon een stelsel evenwijdige lijnen krijgen. De cirkels zijn bepaald door de functie

$$f(x,y) = x^2 + y^2$$

Verschuiving van dat rooster over een kleine afstand δ naar rechts geeft als functie van het tweede rooster

$$f(x + \delta,y) = (x + \delta)^2 + y^2$$

Volgens de hoofdstelling is het patroon bepaald door het verschil

$$f(x + \delta,y) - f(x,y) = 2\delta x + \delta^2 \approx 2\delta x$$

waarbij we δ^2, een heel klein getal, verwaarloosd hebben. Gelijkstelling met nh geeft

$$2\delta x = nh$$

en dat komt inderdaad neer op een stelsel evenwijdige verticale lijnen met de golflengte $h/(2\delta)$. Hoe minder de cirkelschijven ten opzichte van elkaar zijn verschoven des te groter worden de afstanden tussen de moiré-lijnen.

Wie enige kennis heeft van de principes van de differentiaalrekening kan het volgende constateren. Het verschil $f(x+\delta,y) - f(x,y)$ doet ons denken aan de vorming van de afgeleide van een functie. Vergeten we de variabele y even dan wordt immers de afgeleide van $f(x)$ naar x gedefinieerd als de limiet van $(f(x + \delta) - f(x))/\delta$ wanneer δ naar nul gaat. Kenners weten dat hiet sprake is van partieel differentiëren, differentiëren naar x terwijl y constant blijft. Het moiré-patroon van twee identieke roosters die ten opzichte van elkaar een beetje zijn verschoven is dan zoiets als de afgeleide van de hoogtefunctie in de richting van de verschuiving (voor de x-richting dus differentiëren naar x waarbij y constant wordt gehouden).

Het vormen van moiré-patronen komt zodoende neer op grafisch differentiëren.

Het begrijpen van het moiré-patroon van figuur 4.5 is – voor wie over die extra kennis beschikt – niet moeilijk meer. Hij dient de afgeleide van $\sqrt{(x^2+y^2)}$ naar x gelijk te stellen aan nh. De rest is dan eenvoudig.

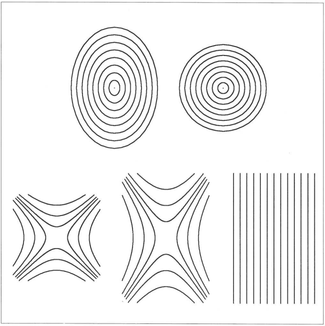
Kegelsneden als moiré

Omdat een moiré-patroon opgevat kan worden als de hoogtekaart van een wiskundig landschap $z = f(x,y)$ kunnen we langs wiskundige weg een systematisch overzicht maken van wat we alzo kunnen waarnemen, hetzij in de textielwereld hetzij in de wereld van de computer. We kunnen, met andere woorden, een soort staalkaart maken van wat we voor bijzonderheden kunnen tegenkomen. Een wiskundige functie van een berglandschap ziet er in de omgeving van een willekeurig punt, we noemen dat lokaal, uit als een hellend plat vlak, een bergtop, een dal of een bergpas. In wiskundige vorm is dat

$$f(x,y) = ax^2 + bxy + cy^2 + \ldots$$

wanneer de coördinaten zodanig zijn aangepast dat de oorsprong $x = 0$, $y = 0$ het te onderzoeken punt is. De belangrijkste verschillen zitten in het kwadratische gedeelte. De traditionele wiskundige analyse zegt ons dat we maar een paar hoofdtypen hoeven te onderscheiden en die geven we hieronder als voorbeelden met een kleine illustratie aan:

$f(x,y) = x^2 + y^2$ concentrische cirkels



figuur 4.7
De elementaire moiré-patronen

$f(x,y) = x^2 + 2y^2$ concentrische ellipsen

$f(x,y) = x^2 - 2y^2$ hyperbolen

De bouwstenen waaruit de moiré-patronen zijn samengesteld hebben we met behulp hiervan in figuur 4.7 samengevat.

Moiré op het beeldscherm

Zoals we al eerder hebben gezegd kunnen enkelvoudige figuren op het beeldscherm door interferentie met het pixelrooster ook aanleiding geven tot moiré-patronen. We beperken ons hier tot de VGA-modus waarbij het beeldscherm uit 640 ⋆ 480 pixelposities is opgebouwd. Een schuine lijn vertoont dan een aantal kleine verspringingen. Worden er veel lijnen tegelijk op het scherm afgebeeld dan vormen die sprongpunten een secundaire figuur, die men als een moiré-patroon kan opvatten. Het is niet moeilijk aan de hand van eenvoudige programma's dergelijke patronen op te roepen. In beginsel beperke men zich tot bundels van evenwijdige lijnen, lijnen door een punt, en cirkelbogen behorende tot concentrische cirkels. De programma's MOIRES1, MOIRES2, MOIRES3 en MOIRES4 mogen de lezer daarbij op weg helpen.

5 Potentialen en velden

Velen van ons kennen ongetwijfeld die leuke natuurkundige experimenten die ook in de huiskamer vertoond kunnen worden, en waarbij niet veel meer nodig is dan een vel papier, wat ijzervijlsel en magneten. Onzichtbare krachtlijnen konden aldus zichtbaar worden gemaakt. Met een computer tot onze beschikking zijn we in staat dergelijke experimenten veel mooier en uitvoeriger uit te voeren. In het bijzonder richten wij onze aandacht op het elektrische veld dat in een vlak gegenereerd wordt door een aantal puntladingen. Omdat er gerekend moet worden denken we dat vlak voorzien van een Cartesisch coördinatenstelsel. De betekenis van het begrip *potentiaal* zou men kunnen toelichten aan de hand van warmte en temperatuur. Dat het ergens op aarde tien graden is zegt niet zoveel, tenzij we die temperatuur kunnen vergelijken met een overeenkomstige temperatuur op een andere plaats of een andere tijd. Wat warm is in de poolstreken is in de woestijn overdag zeker koud. Wel belangrijk is een temperatuurverschil. Een verschil van temperatuur tussen twee naburige plaatsen kan aanleiding geven tot een warmtestroom, een stroom waarvan de grootte ongeveer evenredig is met het temperatuurverschil. Iets soortgelijks treedt op bij stromingen van lucht. Verschil van luchtdruk geeft aanleiding tot het stromen van lucht, tot wind of misschien tot een storm. Zowel temperatuur als luchtdruk zijn voorbeelden van het begrip potentiaal. Van belang is dat men dat begrip kan meten met een enkel getal volgens een zekere schaal. De temperatuur kan men meten in Celsius of in graden Kelvin, maar ook de schaal van Fahrenheit is nog steeds in gebruik. Luchtdruk kan men meten in bijvoorbeeld millimeters kwik of in baren. Een potentiaal is wat men een *scalaire* grootheid noemt. Een warmtestroom, een vloeistofstroom of gewoon de snelheid van een punt is daarentegen een vectorgrootheid, iets wat een richting en een grootte heeft als een pijl. Vectorgrootheden zijn ingewikkelder dan scalaire grootheden. Het is daarom plezierig dat men een aantal berekeningen waarin vectoren optreden kan reduceren tot berekeningen met potentialen.

Het principe waarop dit berust is eigenlijk heel eenvoudig. Is $f(P)$ de waarde van de potentiaal in een punt P en is Q een naburig punt dat op de afstand h van P ligt, dan is er een vectorgrootheid $\mathbf{v}$ die voldoet aan

$$\mathbf{v} \approx c(f(P) - f(Q))/h$$

waarbij c een evenredigheidsfactor is. De formule is des te nauwkeuriger naarmate Q dichter bij P ligt. Het rechterlid kunnen we dan opvatten als een differentiaalquotiënt, maar op de technische bijzonderheden gaan we niet in.

De vector $\mathbf{v}$ heeft de richting van het lijnstukje PQ. Wanneer f de bete-

kenis heeft van een elektrische potentiaal, heeft **v** de betekenis van een elektrische stroom, die net als bij vloeistof in een bepaalde richting stroomt.

Een elektrische puntbron in de ruimte

Aan de elektriciteitstheorie ontlenen we de wetenschap dat een puntlading geplaatst in de oorsprong O van een ruimtelijk coördinatenstelsel een elektrisch veld genereert waarbij de potentiaal in een willekeurig punt P gelijk is aan a/OP, waarbij a een evenredigheidsfactor is die afhankelijk is van de gekozen eenheden. Ter vereenvoudiging kunnen we $a = 1$ nemen. Dus:

De ruimtelijke potentiaal in het punt P(x,y,z)
ten gevolge van een puntbron in de oorsprong O
is $f(P) = 1/OP$ *of* $f(x,y,z) = 1/\sqrt{(x^2 + y^2 + z^2)}$

Aan de puntbron kunnen we een intensiteitsfactor toevoegen die de betekenis van lading heeft. Het teken van die factor correspondeert daarbij met het teken van de lading. Een bron met een negatieve lading pleegt men vaak te betitelen als een *put.* De ruimte kunnen we in gedachten verdelen in vlakken waar de potentiaal telkens constant is. Dat zijn hier boloppervlakken met O als middelpunt. Dergelijke vlakken waarop de potentiaal een zelfde waarde aanneemt heten *potentiaalvlakken.* Van de oorsprong naar buiten gerekend neemt de potentiaal van positief oneindig groot af tot nul. Met het potentiaalveld is onverbrekelijk een vectorveld verbonden, het veld van de elektrische veldsterkte. De veldsterkte kunnen we in gedachten visualiseren als kleine pijltjes die loodrecht op de potentiaalvlakken staan. De vector van de veldsterkte in een punt P heeft dus de richting van de lijn OP en heeft een grootte die we met de bovenvermelde formule kunnen berekenen. Dat gaat voor het veld van een puntbron in de oorsprong het eenvoudigste door P op de x-as te nemen, dus $P(x,0,0)$. De potentiaal is daar $1/x$ en de grootte van de veldsterkte vinden we door die uitdrukking naar x te differentiëren:

$$v = -\partial(1/x)/\partial x = 1/x^2$$

weer met weglating van een evenredigheidsconstante.

In het algemeen is de potentiaal van een puntlading in de oorsprong met de intensiteit a (dus a eenheden van lading) in een willekeurig punt gelijk aan a/OP en is de grootte van de veldsterkte in een punt P gelijk aan a/OP^2.

Ingewikkelder elektrische velden kunnen we op eenvoudige wijze opbouwen door de potentialen van de samenstellende bestanddelen bij elkaar op te tellen. Zijn er in de ruimte bijvoorbeeld drie puntbronnen in de punten A, B en C waarbij de grootte van lading respectievelijk a, b en c is dan is de potentiaal in een willekeurig punt P gelijk aan

$$a/AP + b/BP + c/CP$$

De berekening van de veldsterkte in een willekeurig punt is dan echter een wat moeilijker wiskundig probleem. Natuurlijk staan de vectoren van de veldsterkte loodrecht op de potentiaalvlakken, maar de potentiaalvlakken hebben ook niet zo'n eenvoudige ruimtelijke vorm. Het karwei kan natuurlijk met behulp van de computer geklaard worden, maar omdat het bewerkingen betreft met ruimtelijke objecten blijft het technisch tamelijk ingewikkeld. Liever beperken wij ons tot velden in twee dimensies.

Tweedimensionale velden

In de wiskundige potentiaaltheorie komt als elementaire bouwsteen naast de ruimtelijke puntbron ook het begrip lijnbron voor. Men moet zich daarbij voorstellen dat een oneindig lange rechte lijn op homogene wijze met lading bedekt is. We stellen ons voor dat die lijn de z-as is van een ruimtelijk coördinatenstelsel. Het daardoor verwekte potentiaalveld hangt dan niet meer af van de z-richting, en de potentiaalvlakken zijn cilinders die de z-as als symmetrieas hebben. De veldlijnen staan loodrecht daarop, lijnen door de z-as evenwijdig aan het x,y-vlak. Omdat de z-richting er niet meer toe doet kunnen we net doen alsof we in twee dimensies, in een enkel vlak, werken. Voor dat vlak kunnen we het x,y-vlak gebruiken. We vergeten dan als het ware dat we begonnen zijn met een ruimtelijke situatie en we spreken voortaan van een tweedimensionale puntbron. De potentiaal in een punt $P(x,y)$ die daarbij behoort is niet meer evenredig met de reciproke afstand van P tot de oorsprong O, maar met een logaritmische functie van OP.

> *De tweedimensionale potentiaal in* P(x,y) *ten gevolge van een puntbron in de oorsprong is*
> $f(P) = -\log(OP)$ *of* $f(x,y) = -\log(x^2 + y^2)/2$

Men lette weer op het teken van $f(P)$. Voor een bron is de potentiaal in de omgeving van de bron positief heel groot. Aan de bron kan men weer een intensiteitsfactor toevoegen waarvan het teken aangeeft of we met een positieve of met een negatieve lading te maken hebben. Een bron met een negatieve lading noemen we ook hier weer een put.

De tweedimensionale potentiaaltheorie heeft talloze toepassingsmogelijkheden. Een interessante en voor ons land belangrijke toepassing betreft de stroming van een vloeistof in een ondergrondse poreuze laag. Dat gaat niet alleen om gas en olie, maar vooral ook om grondwater. De potentiaal heeft dan te maken met de druk die het water door zijn omgeving ondervindt, de druk die andere aardlagen uitoefenen en de druk die door de zwaartekracht wordt uitgeoefend. Begrippen als bron en put kan men dan heel letterlijk nemen, en wat in de elektriciteitstheorie de stroomsterkte was heeft hier inderdaad de betekenis van snelheid waarin water zich in een bepaalde richting verplaatst.

Het analogon van een potentiaalvlak in de ruimte is een potentiaallijn. Wie naar de televisie kijkt om te weten wat voor weer het gaat worden zal wel vertrouwd zijn met de weerkaarten waarop zich lijnen van constante druk bevinden. Dat zijn potentiaallijnen. Omdat lucht stroomt van gebie-

den met hoge druk naar gebieden met lage druk bewegen de luchtdeeltjes zich dwars door de potentiaallijnen heen. Helaas wordt het beeld wat vertroebeld doordat de bewegende luchtmassa's de invloed van het draaien van de aarde ondergaan. Zetten we de aarde in onze fantasie even stil, dan zou de baan van een zich verplaatsend luchtdeeltje, een stroomlijn, de potentiaallijnen loodrecht doorkruisen. Op kleinere schaal hebben we in de natuur van die draaiing minder last. Water dat van de bergen naar beneden stroomt kiest op zijn weg steeds het steilste verval. De hoogte kan nu geïnterpreteerd worden als potentiaal, de hoogtelijnen zijn potentiaallijnen, en de stroomlijnen doorkruisen op de hoogtekaart de potentiaallijnen loodrecht.

We bespreken nu een paar simulaties waarin we eenvoudige vlakke potentiaalvelden bestuderen. De fysische betekenis laten we aan de lezer over. Het meest beeldend is misschien een interpretatie als een heuvellandschap, waarover waterdeeltjes onder invloed van de zwaartekracht bewegen. Hoogteverschillen zijn dan potentiaalverschillen, potentiaallijnen zijn lijnen van gelijke hoogte, de banen waarover zich de waterdeeltjes als kleine stroompjes bewegen zijn dan stroomlijnen. In het algemeen zijn van de meeste modellen meer interpretaties mogelijk.

Bron en put

Ons eerste experiment betreft het veld van een eenheidsbron en een eenheidsput. De bron plaatsen we in de oorsprong $O(0,0)$ en de put in het punt $A(a,0)$. De potentiaal in een willekeurig punt P in dit veld is dan gelijk aan

$$f(x,y) = -\log OP + \log AP$$

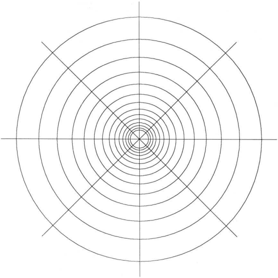
waarvoor we ook

$$f(x,y) = \log(AP/OP)$$

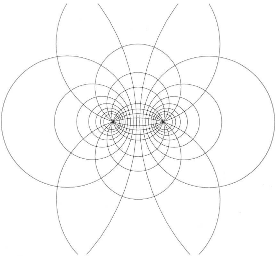
kunnen schrijven.

Wat in de ruimte potentiaalvlakken zijn, zijn hier potentiaallijnen, lijnen waarvoor $f(x,y)$ constant is. Dat betekent dat voor punten P op een potentiaallijn de verhouding van de afstanden van P tot A en van P tot B een constante waarde heeft. Volgens een beroemd klassiek resultaat uit de grote Griekse periode is de potentiaallijn dan een cirkel, een zogenaamde cirkel van Apollonius. De potentiaallijnen vormen met elkaar een zogenaamde cirkelbundel. De stroomlijnen die loodrecht op de potentiaallijnen staan zijn eveneens cirkels, cirkels die door O en door A gaan. Met elkaar vormen ze eveneens een cirkelbundel.

Wanneer we de put steeds verder weg kiezen gaan de stroomlijnen in de omgeving van de bron in de oorsprong steeds meer lijken op rechte lijnen. Wanneer we in gedachten de put naar het oneindige hebben geplaatst blijft alleen de bron in de oorsprong over. De potentiaallijnen zijn dan gewoon concentrische cirkels die de oorsprong tot gemeenschappelijk middelpunt hebben. De stroomlijnen zijn rechte lijnen door de oor-



figuur 5.1 Het potentiaalveld van een puntbron



figuur 5.2 Het potentiaalveld van put en bron

sprong. Voor de bijbehorende potentiaalfunctie kunnen we dan de volgende eenvoudige formule gebruiken:

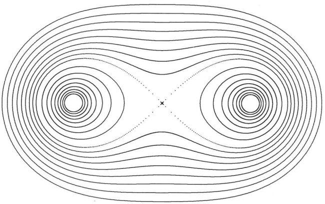
$$f(x,y) = -\log(OP) = -\log(x^2 + y^2)/2$$

In figuur 5.1 hebben we een aantal potentiaallijnen en stroomlijnen afgebeeld die bij een enkele puntbron behoren. Het gebruikte programma is POTVELD1.

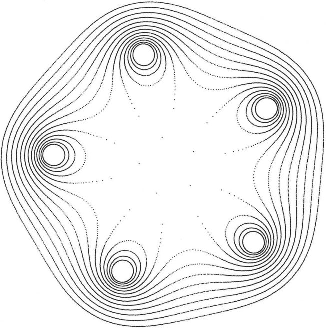
In figuur 5.2 hebben we het potentiaalveld van een bron en een put met gelijke maar tegengestelde lading of intensiteit weergegeven, het bekende patroon van twee elkaar loodrecht doorsnijdende cirkelbundels waarover we in de aanvang van dit hoofdstuk hebben gesproken. Het gebruikte programma is POTVELD2.

Ingewikkelder velden

Er zijn maar heel weinig potentiaalvelden die met eenvoudige wiskundige hulpmiddelen kunnen worden beschreven. Al spoedig komen we in technische problemen, bijvoorbeeld wanneer we het veld beschouwen van twee even grote puntbronnen geplaatst in de punten P_1 en P_2. Op een potentiaallijn is dan telkens het produkt $PP_1 \star PP_2$ constant. Bekend is dat de daardoor gedefinieerde lijnen bekende meetkundige krommen, zogenaamde lemniscaten zijn. Daarvan bestaan twee soorten, kleine ovalen om de bronpunten en grote wat ingedeukte ovalen om beide bronpunten, als geïllustreerd in figuur 5.3. De grenslijn, de potentiaallijn die door de oorsprong gaat, heeft de vorm van een liggende acht. Het is een beroemde figuur die bekend staat als de lemniscaat van Bernoulli. Het tekenen van de lijnen vereist enige wiskundige handigheid zoals het gebruik van poolcoördinaten. We gaan er niet op in, maar we verwijzen



figuur 5.3 Het veld van twee gelijke puntladingen



figuur 5.4
Het veld van een regelmatige vijfhoek

de lezer naar het programma POTVELD3. In dat programma worden de potentiaallijnen weergegeven als stippellijntjes, dat is programmatechnisch het eenvoudigst. Alleen bij het tekenen van de lemniscaat in de omgeving van de oorsprong, waar opvolgende stipjes verder uit elkaar liggen, ziet de tekening er iets minder fraai uit.

Een aardige generalisatie is een veld waarbij op de hoekpunten van een regelmatige veelhoek onderling gelijke puntladingen zijn geplaatst. In figuur 5.4 (zie POTVELD4) geven we daarvan een illustratie. Op de wiskundige behandeling gaan we niet in. Ook daarbij zijn weer wat kleinere tekentechnische problemen in de omgeving van de oorsprong.

Het veld van een condensator

In hoofdstuk 3 hebben we gezien hoe we met behulp van het programma HOOGTE2 een kleurige hoogtekaart konden ontwerpen van een door een willekeurige functie $z = f(x,y)$ bepaald landschap. Dat kan natuurlijk ook een potentiaallandschap zijn. De mogelijkheid bestaat om via een uitgebreider programma een willekeurig potentiaalveld samen te stellen uit een willekeurige combinatie van bronnen en putten. Het volstaat daarin het totale aantal te vermelden met van elke bron (of put) de positie x,y en de intensiteit, positief voor een bron en negatief voor een put. In fi-

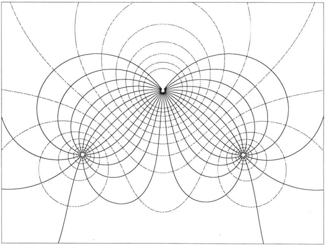
guur 5.5 (zie bladzijde 82 in het kleurkatern) hebben we een serie van bronnen op een lijnstuk gecombineerd met een serie putten op een daaraan evenwijdig lijnstuk. Alle intensiteiten nemen we daarin gelijk, bronnen +1 en putten -1. Aldus kunnen we het veld van een plaatcondensator imiteren, wanneer tenminste het aantal puntbronnen groot genoeg is.

Als illustratie geven we in figuur 5.6 (zie bladzijde 82) een afbeelding van een veld dat opgebouwd is uit twee puntbronnen in (-1,0.5) en (1,0.5) met dezelfde intensiteit en een put in (0,-0.5) van dubbele sterkte.

Een professionele methode

In het derde hoofdstuk hebben we het mooie programma LEVEL besproken waarmee we in staat waren een serie niveaulijnen van een willekeurige landschapsfunctie te tekenen. Dat programma vereist wel geduld van de gebruiker, tenzij hij over een snelle computer beschikt. Het is niet moeilijk om dat programma nog wat uit te breiden op een zodanige wijze dat er een aantal stroomlijnen aan wordt toegevoegd.

Het principe lichten we toe aan de hand van een oppervlak $z = f(x,y)$ waarover we een knikkertje laten rollen, onderworpen aan de in de z-richting werkende zwaartekracht. Het knikkertje kiest telkens op zijn baan de steilste richting naar beneden. Nemen we aan dat het knikkertje zich in de positie (x,y) op de hoogte z bevindt, dan is de omgeving van het knikkertje bij benadering een plat vlakje, een deel van het raakvlak aan het z-oppervlak. Onze taak is het de vector te bepalen waarin het knikkertje beweegt. Wie iets van differentiaalrekening weet kent het antwoord op



figuur 5.7 Potentiaallijnen en stroomlijnen

die vraag als de vector van de partiële afgeleiden van z naar x en y. In gewonere taal is het de vector waarvan de componenten evenredig zijn met de plaatselijke daling in de x-richting en de overeenkomstige daling in de y-richting

$$dx = f(x,y) - f(x+h,y), \qquad dy = f(x,y) - f(x,y+h)$$

waarbij h een klein getalletje is. De grootheden dx en ds kunnen we, na het aanbrengen van een evenredigheidsconstante, interpreteren als de verplaatsing van het punt dat zich langs een stroomlijn over het beeldscherm beweegt. In figuur 5.7 laten we zien hoe het in figuur 5.6 afgebeelde potentiaalveld eruit ziet wanneer er stroomlijnen aan worden toegevoegd.

6 Zelfgelijkvormige plantengroei

Biologen hebben al vaak opgemerkt dat de bouw van levende wezens niet alleen veel regelmatigheid vertoont, maar dat het soms is alsof een zelfde groeiproces zich op verschillende plaatsen en tijden telkens herhaalt. We hoeven maar naar een boom, een vogelveer of een varenblad te kijken om daarvan overtuigd te raken. Een van de grondgedachten van de natuurwetenschap is om te trachten de veelsoortige verschijnselen op zo eenvoudig mogelijke wijze te kunnen begrijpen. We plaatsen ons als het ware op de stoel van de Schepper en we gaan uit van een paar simpele natuurwetten van meetkundige aard. We zijn daarbij in het goede gezelschap van de Griekse filosoof Plato, die reeds verklaarde dat God bij de schepping van de wereld op meetkundige wijze te werk was gegaan. Om een idee van de mogelijkheden te krijgen en in een streven om alles zo eenvoudig mogelijk te houden beperken we ons in dit hoofdstuk tot een wereld van twee dimensies.

De meetkundige simulatie steunt op het begrip van de *gelijkvormigheidstransformatie*. Voor het platte vlak wil dat zeggen dat we het vlak in zijn totaliteit onderwerpen aan een verplaatsing en een inkrimping. Een zich in het vlak bevindende rechthoek zal ten gevolge van een dergelijke behandeling ergens anders, verkleind, soms gedraaid en vervormd, terechtkomen. Een verplaatsing kunnen we ons het beste voorstellen door te beginnen met een vel papier waarop bijvoorbeeld een vierkant is getekend. Daarop leggen we een transparant vel, en we kopiëren daarop het vierkant. Het doorzichtige vel verschuiven we en tegelijk draaien we het wat. We kunnen nu goed het verplaatste vierkant vergelijken met het origineel in het onderste vel. Het bovenste vel kunnen we ook in zijn geheel omdraaien en weer in een andere positie op het onderste papier leggen. We hebben dan te maken met een spiegeling. Voor een vierkant maakt dat niets uit, omdat een vierkant zelf spiegelsymmetrisch is, maar zouden we als basisfiguur bijvoorbeeld de letter F hebben genomen, dan is er natuurlijk wel verschil. Om het effect van een inkrimping te zien kunnen we het beste beginnen met het aanbrengen van een verkleining van het origineel in het bovenste transparante vlak om het vervolgens door verplaatsing in een willekeurige positie te brengen.

In dit hoofdstuk beperken we ons tot zogenaamde lineaire of, meer speciaal, affiene transformaties. Zij hebben als kenmerk dat rechte lijnen altijd recht blijven, en dat een willekeurig vierkant getransformeerd wordt in een parallellogram. In coördinaten, onmisbaar bij het opstellen van een computerprogramma, ziet zo'n transformatie eruit als

$$(6.1) \qquad \begin{aligned} x' &= ax + by + e \\ y' &= cx + dy + f \end{aligned}$$

De getallen a, b, c, d hebben te maken met de vormverandering van bijvoorbeeld een willekeurige rechthoek. Het oppervlak van de rechthoek ondergaat daarbij een verandering met de constante factor ad-bc, een belangrijk getal. Is dat getal negatief dan heeft de transformatie het karakter van een spiegeling, is dat getal positief dan is er een gewone vervorming. Is ten slotte $ad - bc = 0$ dan is de transformatie een *projectie* waarbij alles tot een enkele rechte lijn wordt samengedrukt.

Bij een aantal concrete voorbeelden heeft men te maken met een paar speciale gevallen van (6.1) die elders uitvoerig besproken zijn, maar waarover we hier toch iets moeten zeggen. Wanneer bij een gelijkvormigheidstransformatie als (6.1) of iets met ingewikkelder formules een punt op zijn plaats blijft noemen we dat punt een *dekpunt*. Zo'n punt kunnen we berekenen door uit te gaan van $x' = x$ en $y' = y$. Een draaiing van een vlak om de oorsprong (0,0) over een hoek waarbij tegelijk de afstanden tot de oorsprong verkort worden met de factor s ziet er in formulevorm uit als

$$x' = s(x \cos \alpha - y \sin \alpha) \qquad (6.2)$$
$$y' = s(x \sin \alpha + y \cos \alpha)$$

De oorsprong (0,0) blijft op zijn plaats en is dus een dekpunt. Is in het bijzonder $\alpha = 0$ dan reduceert (6.2) zich tot

$$x' = sx, \qquad y' = sy$$

de bekende centrale vermenigvuldiging. De factor s is de eendimensionale verkortingsfactor van lijnstukken, wel te onderscheiden van de tweedimensionale verkortingsfactor van rechthoeken en dergelijke die gelijk is aan s^2.

De boom van Pythagoras

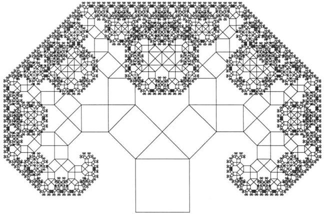
Het althans in Nederland meest bekende geometrische model van zelfgelijkvormige plantengroei is de beroemde boom van Pythagoras, ontstaan in de Tweede Wereldoorlog op de tekentafel van ir. A. Bosman en afgebeeld in figuur 6.1.

Na enig speurwerk kan men ontdekken dat de hele figuur zelfgelijkvormig is ten opzichte van twee verschillende gelijkvormigheidstransformaties van het door (6.2) beschreven type. De eerste, aangeduid door L, is een draaiing over een hoek van 45 graden gecombineerd met een lineaire verkorting van $s = 1/\sqrt{2}$ waarvan het dekpunt zich links beneden bevindt als geschetst in figuur 6.2. De tweede, symbolisch R genoemd, is een overeenkomstige draaiing maar dan in tegengestelde zin. Het dekpunt bevindt zich rechts beneden. Door de werking van L of R wordt elk onderdeel van de Pythagoras-boom verplaatst, verkleind en neemt dan de plaats in van een ander onderdeel van de boom. De stam wordt een tak waaruit als een stam een nieuwe boom ontstaat.

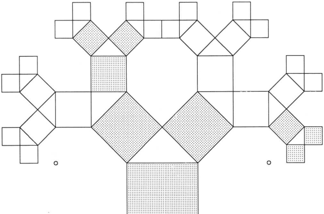
Elk kleiner vierkant van de boom van Pythagoras kan aldus uit de abscis, het grootste vierkant, als stam worden afgeleid door maar vaak ge-

noeg in een bepaalde volgorde de transformaties L en R erop los te laten. Heel mooi wordt het wanneer we de kruin van de boom bekijken, de kleinste vierkantjes. Dat zijn er theoretisch oneindig veel en het is natuurlijk onbegonnen werk om ze allemaal te willen tekenen, ook al zouden we ons om praktische redenen tot heel kleine afmetingen willen beperken. We doen dat in een computersimulatie liever op 'chaotische' wijze als een 'spel van chaos en orde'. We nemen aan dat we ons bevinden in de positie *P* in een van die allerkleinste vierkanten. Die zijn dan elk tot een punt ineengeschrompeld. Het punt *P* kunnen we dan zien als de meetkundige limiet van een rij krimpende vierkantjes. We maken nu gebruik van de

figuur 6.1
De boom van Pythagoras



figuur 6.2
Zelfgelijkvormigheid in de boom van Pythagoras



wetenschap dat zowel L als R dat punt P in een ander punt van de kruin overvoert. Elk punt P van de kruin levert dus meteen twee andere punten L(P) en R(P) van de kruin. We kiezen op willekeurige wijze een van de twee uit en daarmee gaan we verder op precies dezelfde wijze, telkens weer dobbelen om een volgende keuze te maken. Op het beeldscherm zien we aldus een rij van punten P, P_1, P_2, P_3, ... ontstaan waarbij elk punt uit het voorafgaande is afgeleid door er de transformatie L of R op los te laten. In een computerprogramma is zoiets gemakkelijk uit te voeren en met behulp van het programma BOOMMC is dan figuur 6.3 gevormd.

Dat programma is een typisch voorbeeld van de zogenaamde Monte-Carlo-methode, naar het bekende oord waar men pleegt te gokken. Inderdaad wordt in dit programma telkens een stochastische keuze gemaakt, de officiële wiskundige benaming voor een gokje, door gebruik te maken van de Random Number generator van de computer die toevallige getallen tussen 0 en 1 produceert. Bij de kruin van de boom van Pythagoras verwerken we die toevallige getallen op een kruis-of-munt-manier omdat de transformaties L en R ten aanzien van contractie gelijkwaardig zijn. Het programma maakt gebruik van de rotatieformules (6.2) waarbij

$$a = s \cos \alpha, \qquad b = s \sin \alpha$$

voor de eerste transformatie L, terwijl c, d een overeenkomstige betekenis hebben voor de tweede transformatie R. Het programma werkt net zo goed voor rotaties met andere parameters. Goede keuzen zijn bijvoorbeeld:

$a = 0.6$	$b = 0.6$	$c = 0.48$	$d = 0$
$a = 0.6$	$b = 0.45$	$c = 0.5$	$d = 0$
$a = 0.75$	$b = 0.25$	$c = 0.5$	$d = -0.375$

Wanneer de contractiewaarden van de twee transformaties verschillen kan op het beeldscherm een onevenwichtig resultaat zichtbaar worden met kale en met overvolle gedeelten. Het is dan beter de keuze van L of R af te stemmen op de tweedimensionale contractiefactoren die gelijk zijn aan $a^2 + b^2$ voor L en $c^2 + d^2$ voor R. Is de eerste factor bijvoorbeeld gelijk aan 0.5 en de tweede gelijk aan 0.25 dan wil dat zeggen dat we L tweemaal zo vaak als R moeten kiezen.

Met een kleine aanpassing is het programma ook bruikbaar voor bijvoorbeeld twee gespiegelde gelijkvormigheidstransformaties als gegeven door

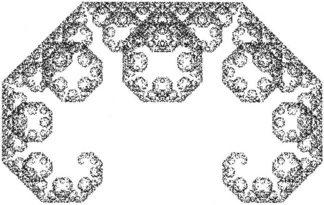
$$x' = ax + by - 1 + a, \qquad y' = bx - ay + b$$
$$x' = cx + dy + 1 - c, \qquad y' = dx - ay - d$$

Goede keuzen zijn dan

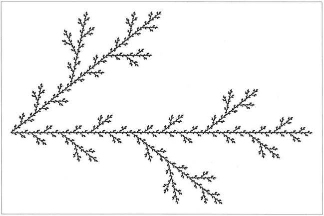
$a = 0.5$	$b = 0.5$	$c = 0.5$	$d = 0$
$a = 0.5$	$b = 0.5$	$c = 0.667$	$d = 0$

 We krijgen dan figuur 6.4.

figuur 6.3
De kruin
van de boom
van Pythagoras



figuur 6.4 Een
zelfgelijkvormig
boomtakje

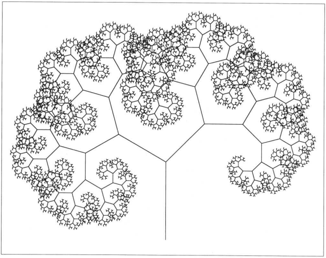


Nog meer bomen

Fantasierijke computeraars die mogelijk inspiratie hebben opgedaan bij een boswandeling hebben talloze varianten van de Pythagoras-boom bedacht. Ook al omdat er elders in de literatuur veel aandacht aan is besteed beperken we ons hier tot het enkele voorbeeld van een 'kale' boom met lijnvormige stam en takken die in een bepaald ritme steeds kleiner worden. Het basisschema is aangegeven in figuur 6.5. De stam, en vervolgens elke tak, vertakt zich naar links onder een hoek α en een verkortingsfactor a en naar rechts met de hoek β en de contractiewaarde b.

Met behulp van het programma BOOMBT kan dan voor de waarden

figuur 6.5
Het basispatroon van een zelfgelijkvormige boom



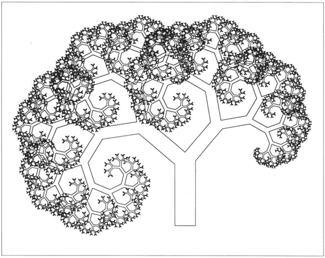
$a = b = 0.7$, $\alpha = \pi/3$ en $\beta = \pi/4$ de in figuur 6.5 afgebeelde boom worden verkregen. Het programma maakt gebruik van de weinig geheugenruimte vergende methode van backtracking en de zogenaamde techniek van de stapel. In het vorige hoofdstuk hebben we al met deze methode kennis gemaakt en ook in het volgende hoofdstuk zal die methode aan de orde komen. Het programma gaat door met het vormen van takjes totdat alle mogelijkheden met een taklengte boven een vooraf gegeven drempel uitgeput zijn. Het programma is zeer flexibel en laat een groot aantal varianten toe (zie bijvoorbeeld figuur 6.6). Men zij er evenwel op bedacht dat de WINDOW-opdracht aangepast moet worden wanneer de boom te groot of te klein dreigt te worden.

De boom- en plantachtige figuren die als gevolg van het 'spel van chaos en orde' ontstaan noemt men tegenwoordig fractals (of fractalen). Het algemene kenmerk is hun zogenaamde zelfgelijkvormigheid. Bij de kruin van de Pythagoras-boom, het resultaat van twee contracties L en R, gaat de hele figuur op gelijkvormige wijze over in een deel van de figuur.

Het varenblad van Barnsley

In de geschiedenis van de fractalmeetkunde plachten fractals meestal een min of meer toevallig nevenprodukt te zijn van een wiskundige theorie. Stellig was dat het geval bij de door Gaston Julia omstreeks 1918 gevonden objecten die te maken hadden met de iteratie van eenvoudige wiskundige functies als veeltermen. Zoals we hierboven hebben gezien wordt door een paar gelijkvormigheidstransformaties een fractal bepaald, maar wanneer we de formules zien hebben we doorgaans geen idee hoe de frac-

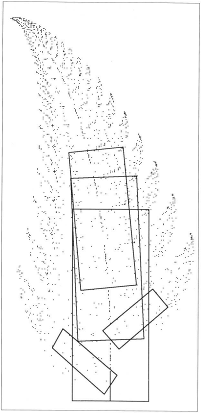
figuur 6.6 Een variant van de Pythagoras-boom



tal eruit zal zien wanneer we hem op het computerscherm zichtbaar kunnen maken. Het boomtakje van figuur 6.4 is dan het resultaat van een gelukkige keuze van parameterwaarden. De wiskundige Michael Barnsley heeft zich met succes met het omgekeerde probleem beziggehouden. Hoe kunnen we een bepaalde vorm als een boomblad of een vogelveer, een object waarin we een zekere zelfgelijkvormigheid kunnen waarnemen, als fractal genereren met behulp van een aantal contracties, samentrekkende gelijkvormigheidstransformaties? In zijn boek *Fractals Everywhere* en in zijn publikaties spreekt hij over 'fractals op bestelling'. Zijn werkwijze kan goed geïllustreerd worden aan de hand van een varenblad, een van zijn meest succesvolle illustraties. Het varenblad zelf is afgebeeld in figuur 6.7.

In figuur 6.8 zien we hoe het varenblad als een spel van chaos en orde kan worden samengesteld uit vier gelijkvormigheidstransformaties. We gaan daarbij uit van een smalle rechthoek geplaatst aan het begin van het blad, symmetrisch ten opzichte van de hoofdnerf. Twee van de transformaties maken daar twee kleinere rechthoeken van als oervormen van de onderste twee zijblaadjes. Een derde transformatie is verantwoordelijk voor het opschuiven en kleiner worden van de blaadjes naar het einde toe. De combinaties van een van de eerste twee transformaties met een herhaling van de derde contractie zorgen voor de beginfase van de overige aan de hoofdnerf gehechte zijblaadjes. Een combinatie van de transformaties waarin een van de eerste twee contracties wordt herhaald levert de secundaire en hogere bladvormen. Als een finesse voegen we er een steelvorm aan toe door er nog een vierde transformatie bij te betrekken, een projectie waarin het gehele blad tot een stukje steel of nerf wordt samengedrukt. In het programma VAREN kan men het nagaan.

figuur 6.7 Het varenblad van Barnsley



figuur 6.8 De samenstellende contracties van het varenblad van Barnsley

7 Wiskunde en wiskunst

Over kunst kunnen we eindeloos discussiëren, al is het alleen maar over wat het begrip kunst eigenlijk inhoudt. Een eensluidende definitie is er niet, maar we zijn het er wel over eens dat het iets te maken heeft met het overdragen van emoties, van kunstenaar tot publiek. Een kunstenaar dient daartoe over fantasie en over een zekere techniek te beschikken die bij 'gewone mensen' minder voorkomen. Een kunstenaar zou aldus in staat moeten zijn zijn omgeving voor verrassingen te plaatsen en uitroepen als 'wat mooi' of 'wat knap' uit te lokken. Dat Rembrandt een groot kunstenaar is staat voor iedereen wel vast, dat bijvoorbeeld Piet Mondriaan of Paul Klee een vooraanstaande plaats in de kunstgeschiedenis van de twintigste eeuw innemen zal ook wel aller instemming hebben. Maar over de hedendaagse kunst en over de talloze -ismen, te beginnen met impressionisme, expressionisme en kubisme, lopen de meningen al sterk uiteen. Het is hier niet de plaats om daarin partij te kiezen, maar wel mag gezegd worden dat kunst in alle vormen in onze samenleving geïntegreerd dient te zijn. Kunst moet ook tot ons komen in de vorm van toegepaste kunst, in de architectuur bijvoorbeeld, of als decoratieve kunst, zeg ter versiering van ons milieu.

Wiskunde is een exacte wetenschap die zich bezighoudt met een nauwkeurig omschreven begrippenwereld, de wereld van punt en getal. Wiskunde ergens in de kosmos, op een planeet van Sirius, is niet anders dan onze wiskunde. Maar de kunst van de 'Siriërs', een spel van kosmische stralen misschien, valt vermoedelijk buiten ons bevattingsvermogen. Wiskunde en kunst lijken elkaars tegenpolen, als tegengestelde componenten in een binaire wereld van Yin en Yang, van alfa en van bèta, van goed en van kwaad.

Kunst en wiskunde hebben meer gemeen dan men oppervlakkig zou denken. Voor beide disciplines is veel fantasie nodig. Een goed gevoel voor ruimtelijke vormen is van belang voor beeldhouwer en meetkundige. De modellenkast van een mathematisch instituut bevat objecten die in een museum van moderne kunst niet zouden misstaan, en ook het omgekeerde is waar. Grote kunstenaars van de Renaissance als Piero della Francesca waren in hun tijd ook wiskundigen die pionierswerk verrichtten op het gebied van de perspectief.

Grafische kunstenaars hebben altijd op dankbare wijze gebruik gemaakt van technische hulpmiddelen om hun ideeën gestalte te geven. Maar toch moet men veel tijd kwijt zijn geweest aan het snijden van houtblokken, het graveren van koperen of stalen platen en aan het maken van afdrukken. Bekend is dat Japanse prentkunstenaars als Hokusai en Hiroshige die technische rompslomp aan anderen overlieten. Zij beperkten zich tot het maken van tekeningen en schetsen en lieten de uit-

werking aan anderen over. Aldus konden dergelijke kunstenaars tot een gigantische produktie van enige tienduizenden prenten komen.

De computer als grafisch gereedschap

De moderne graficus heeft met de computer een stuk gereedschap in handen gekregen waarvan we de mogelijkheden nog maar nauwelijks zijn gaan bevatten. Daar komt natuurlijk wel een heel nieuwe techniek voor kijken, een techniek die berust op de beschikbaarheid van goede apparatuur en goede programmatuur. Er bestaan tegenwoordig een aantal min of meer professionele softwarepakketten ten dienste van grafisch ontwerpers in de sfeer van desktop-publishing. Dat is allemaal heel mooi, maar ligt toch wel buiten het bereik van de gemiddelde computergebruiker. In dit hoofdstuk zullen we laten zien hoe men gewapend met enige – bescheiden – wiskundige kennis en met gebruikmaking van de mogelijkheden van een moderne programmeertaal – een moderne versie van Basic bijvoorbeeld – tot fraaie decoratieve resultaten kan komen. Wanneer we in onze programma's de namen van kunstenaars als Mondriaan en Klee noemen, gaat het daarbij niet zozeer om een simulatie, maar meer om een associatie. Afgezien van het 'idee', de creatieve kern van de programma's, is het vaak het element van toeval waaraan het resultaat zijn ontstaan te danken heeft. Hoe het toeval in onze programma's is verwerkt kunnen we het beste aan concrete voorbeelden zien.

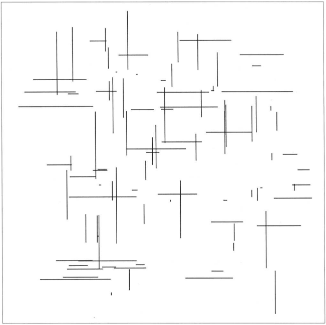
Grafisch ontwerpen vereist wel een redelijk moderne computer, zeg een MS-DOS-computer met ten minste een 80286-processor met zestien kleuren in de EGA/VGA-modus. In de preferabele VGA-modus heeft het beeldscherm dan een oplossend vermogen van 640 ⋆ 480 pixels. Wat we gemakshalve maar computerkunst zullen noemen, het maken van prettig aandoende patronen of figuren, kunnen we ook monochroom, als zwart en wit, uitvoeren. In werkelijkheid kunnen dat ook twee andere kleuren zijn, afhankelijk van de apparatuur en van de voorkeur van de gebruiker. Overigens zijn voor vele grafische ontwerpen kleuren niet altijd nodig, en komen de grafische kwaliteiten van een tekening vaak sterker naar voren naarmate het aantal kleuren minder is. Het is daarom verstandig om in een grafisch programma het aantal beschikbare kleuren a priori te kunnen kiezen. Wanneer kleuren gewenst zijn is het ook prettig om in een reeds op het beeldscherm gevormde figuur de kleuren te kunnen veranderen. Dat is met behulp van slechts enkele opdrachten heel goed mogelijk, zoals uit de bij dit hoofdstuk behorende programma's moge blijken.

Kunst op de wijze van Mondriaan

We beginnen met het programma MONDRIAN waarin nog geen kleuren worden gebruikt. In dit programma wordt binnen een vierkant een voorgeschreven aantal horizontale of verticale lijnstukjes getekend waarbij plaats en grootte volgens toeval bepaald zijn. De opdracht RANDOMIZE #, waarbij voor # een willekeurig geheel getal gekozen kan worden,

legt het begin vast van een reeks toevallige getallen RND, getallen tussen 0 en 1. Dezelfde keuze van dat getal, de zogenaamde 'random seed', leidt altijd tot dezelfde reeks toevallige getallen en dus tot hetzelfde plaatje. Die herhaalbaarheid kan in de praktijk gemakkelijk zijn wanneer we een bevredigend resultaat willen bewaren. Wie bij herhaling van een zelfde programma telkens iets anders wil zien, kan de opdracht RANDOMIZE TIMER gebruiken. In feite is de 'seed' dan door de computerklok bepaald. Alle uitkomsten van RND zijn even waarschijnlijk. Met de toevoeging van een functie f die getallen van het interval (0,1) omzet in andere getallen van hetzelfde interval kunnen andere waarschijnlijkheidsverdelingen worden gemaakt. Hier is de lengte van elk toevallig lijnstukje afgeleid van √(RND). De waarschijnlijkheidsverdeling van √RND is asymmetrisch ten gunste van de kleine waarden. In het programma kunnen het aantal lijntjes N en de 'gemiddelde' lengte H nader gekozen worden. Een resultaat, dat ons doet denken aan de kunst van Mondriaan, zien we in figuur 7.1.

Het programma ARTBLOK laat zien hoe op een meer professionele wijze met kleuren kan worden gewerkt. Het uitgangspunt is een schaakbord waarvan de velden op toevallige wijze gekleurd kunnen worden. Maar in het programma gaat het veel algemener. Het aantal velden kan



figuur 7.1 Een patroon als van Mondriaan

worden gekozen als $m \star m$ door de parameter m aan het begin een bepaalde waarde te geven. De grootte van het gekleurde vierkantje kan met de parameter a worden geregeld, $a = 1$ is vullend, $a = 0.8$ is een iets kleiner gekleurd blokje in het beschikbare veld. Het aantal kleuren hangt af van de keuze van numcol, en de vullingsgraad hangt af van q. De waarde $q = 0.7$ betekent dat ongeveer zeventig procent van de velden gekleurd wordt.

Wanneer op het scherm het patroon verschijnt kan met een willekeurige toetsaanslag een andere kleuring worden verkregen. Dat kan eindeloos worden herhaald, en slechts met de 'X'-toets kan het programma worden gestopt. Het geheim van deze fraaie truc zit verstopt in de subroutine 'othercolours', in principe het gebruik van de palette-opdracht. Men zij er evenwel op bedacht dat Turbo Basic en Q(uick)Basic zich ten aanzien van de palette-opdracht in de VGA-modus verschillend gedragen. Zie figuur 7.2, bladzijde 83 in het kleurkatern.

Kunst op de manier van Klee

De kunstenaar Paul Klee (1879-1940) heeft een aantal bizarre voorstellingen gemaakt waarbij bepaalde elementaire figuren op schijnbaar toevallige manier geordend lijken. Sommige van zijn scheppingen doen denken aan kindertekeningen, maar dan moeten het wel heel artistiek begaafde kinderen zijn geweest. In het programma KLEE betuigen wij hommage aan deze Duits-Zwitserse kunstenaar (figuur 7.3, bladzijde 84).

Het programma is verwant aan MONDRIAN, maar in plaats van horizontale en verticale lijnstukjes worden nu op willekeurige plaatsen gekleurde lijnstukjes of driehoekjes in een willekeurige oriëntatie en van een door toeval bepaalde grootte geplaatst. Het programma is zodanig geschreven dat men er zelf uitbreidingen of veranderingen aan kan geven. Men zou bijvoorbeeld de subroutine 'lijn' kunnen vervangen door een subroutine 'cirkel' met behulp waarvan gekleurde cirkelschijfjes kunnen worden gemaakt, bijvoorbeeld

```
C=1+INT(4*RND)
CIRCLE (X,Y),V,C : PAINT (X,Y),C
RETURN
```

Eieren

Oude volksgebruiken dreigen te verdwijnen omdat onze moderne maatschappij zo sterk op consumptie van in overvloed aanwezige massawaar is gericht. Toegegeven moet worden dat maar weinigen over artistieke vermogens of handvaardigheid beschikken, of denken te beschikken, om zelf paaseieren te beschilderen. In elk geval zijn er vele mogelijkheden om op het beeldscherm patroontjes te ontwerpen die kunnen dienen als voorbeeld van beschildering van een paasei. Het eerste probleem dat zich aandient is de wiskundige simulatie van een ei.

Een ei is wiskundig gesproken een omwentelingslichaam. Doorsneden loodrecht op de omwentelingsas zijn altijd cirkels. Een doorsnede die de omwentelingsas bevat, en dus de karakteristieke eivorm heeft, ziet er altijd eender uit. Het is voldoende wanneer we zo'n doorsnede bekijken. We gebruiken een x,y-coördinatenstelsel waarbij we de x-as als omwentelingsas kiezen. Die as is dus een symmetrieas van de figuur. Het eenvoudigste ei is een zuivere bol, en ook om biologische redenen is dat de beste vorm, omdat een bol bij een gegeven inhoud het kleinste oppervlak heeft. Maar omdat een ei in de natuur door een cilindrische buis moet worden getransporteerd zijn eieren van nature enigszins cilindrisch samengedrukt, en hebben ze een puntige en een meer stompe top.

De cirkelvorm is wiskundig bepaald door

$$y^2 = 1 - x^2 \qquad \text{of} \qquad y = \pm\sqrt{(1 - x^2)}$$

Om een afplatting te bereiken is het voldoende om de y-waarden evenredig te verkleinen, $y \rightarrow ay$ waarbij $a < 1$. We krijgen dan de functionele beschrijving

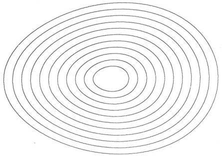
$$y = \pm a\sqrt{(1 - x^2)}$$

waarvan de grafiek een ellips is met toppen in $(\pm 1,0)$ en $(0,\pm a)$. Ten slotte moet er nog een kleine aanpassing komen die de verschillende topvormen van een ei kan beschrijven. Daartoe voegen we aan de functie onder het wortelteken nog een factor $1+bx$ toe waarbij b een klein positief getal is. Omdat het effect van die factor afhangt van het teken van x is er zodoende een beetje asymmetrie aangebracht. We proberen dus of de nieuwe functie

$$y = \pm a\sqrt{(1 - x^2)(1 + bx)}$$

een bevredigend resultaat geeft. In figuur 7.4 zien we dat de eivorm heel goed benaderd wordt. De simulaties kunnen met het programma EI worden uitgevoerd door in het programma de waarden van a en van b telkens te veranderen.

figuur 7.4
Een nest van wiskundige eieren



Een geblokt ei

Er zijn verschillende methoden om een gekleurd ei, zeg een paasei, op het beeldscherm te krijgen. Een eenvoudige manier is om uit te gaan van een vierkant dat op de een of andere wijze van een decoratie of kleuring is voorzien en om dat vierkant met decoratie en al te transformeren in een eivorm. Ter wille van de eenvoud beperken we ons daarbij tot een ei dat de vorm heeft van een zuivere ellipsoïde met doorsnede

$$x^2 + (y/b)^2 = 1$$

Beginnen we met een versierd eenheidsvierkant $-1 < x < 1, -1 < y < 1$ dan kan dat vierkant punt voor punt in een ellipsvormig gebied getransformeerd worden met de formules

$$x' = x, \qquad y' = by\sqrt{(1 - x^2)}$$

We krijgen aldus fraaie ontwerpen in een elliptisch gebied. Bij deze methode is geen rekening gehouden met de driedimensionale vorm van het ei en met het gegeven dat we driedimensionale objecten in een tweedimensionale projectie waarnemen. Wil men dat de figuur op het beeldscherm er echt uitziet als de projectie van een ruimtelijk ei, dan moeten de verticale afstanden nog worden aangepast. Een doelmatige verbetering is dan

$$x' = x, \qquad y' = b\sqrt{(1 - x^2)}\,\sin(\pi y/2)$$

Een resultaat verkregen met het programma BLOKEI is afgebeeld in figuur 7.5 (zie bladzijde 85).

De structuur van het programma is eenvoudig, maar er is door het gebruik van kleuren wel een technische complicatie. Wanneer we op het beeldscherm een gebied willen kleuren kunnen we de opdracht

```
PAINT (X,Y),C
```

gebruiken waarbij de door het attribuut C bepaalde kleur wordt gekozen. Uitgaande van het punt (x,y) worden alle naburige pixels van dezelfde kleur voorzien totdat een gesloten grenskromme met dezelfde kleur wordt ontmoet. Is een dergelijke kromme niet aanwezig of is door een kleine onnauwkeurigheid de kromme niet helemaal gesloten, een beetje lek, dan strekt de kleuring zich tot het gehele beeldscherm uit, wat natuurlijk niet bedoeld was. Om een cirkelvlakje te kleuren moet dus eerst de cirkel in de gewenste kleur worden getekend voordat de PAINT-opdracht uitgevoerd kan worden. In die opdracht kunnen we voor (x,y) natuurlijk het middelpunt nemen. De kleuring van een rechthoekje is al even eenvoudig. Maar hier zijn de te kleuren blokjes omgeven door ellipsvormige bogen en die moeten met veel zorg uit een aaneensluitende rij van pixels worden opgebouwd. En dat is de verklaring voor de uitgebreide 'trafo' waarin die beranding wordt uitgevoerd.

Een ei als dynamisch systeem

Kunst berust ook op een voortdurende waarneming waarin men alert is op toevalligheden. Iets als 'art trouvé'. In het wetenschappelijke bedrijf stuit de onderzoeker vaak op objecten waarop, met een knipoog, het etiket 'kunst' van toepassing kan zijn. Hebben niet reeds fraaie series van gekleurde Julia-fractals en details van de Mandelbrot set musea afgereisd? Zo kunnen experimenten met iteratieve dynamische systemen wel eens een eivorm tot aantrekker hebben. Het principe is de vorming van een puntenrij volgens een functioneel voorschrift als

$$x_{n+1} = f(x_n, y_n), \qquad y_{n+1} = g(x_n, y_n)$$

waarbij n de rij van de natuurlijke getallen doorloopt.

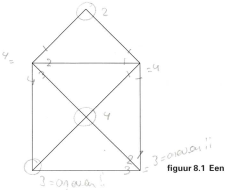
Vaak gaan de punten van een dergelijke baan naar een limietfiguur toe, een zogenaamde aantrekker, die kan bestaan uit een of meer punten, een kromme of iets heel ingewikkelds. Vaak ziet zo'n aantrekkende kromme eruit als een ellips of een ovaal. Dat is het geval in het programma PAASEI waarin naburige banen op spiraliserende wijze naar een eivormige kromme toegaan. In plaats van punten hebben we kleine gekleurde cirkelschijfjes getekend. Bovendien is er de voorziening in aangebracht om in het verkregen resultaat de kleuren te kunnen veranderen. Een heel bijzonder effect kan worden verkregen door een willekeurige toets, de spatiebalk bijvoorbeeld, ingedrukt te houden. Er ontstaat dan iets als een wervelende beweging. Het programma bevat als gebruikelijk een aantal parameters waarvan men de waarde op voorzichtige wijze zou kunnen variëren (figuur 7.6, bladzijde 86).

8 Het pad van de schildpad

Menigeen herinnert zich uit zijn kinderjaren een opgave als 'teken de volgende figuur zonder pen of potlood van het papier te halen'. Het ging dan vaak om een figuur die bestond uit een aantal onderling door lijnen verbonden punten. Een eenvoudig voorbeeld is het zogenaamde huisje, afgebeeld in figuur 8.1.

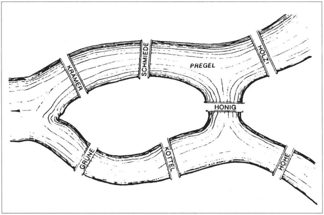
Wie zich later met grafentheorie, een traditioneel onderdeel van modern wiskundeonderwijs, ging bezighouden weet dat het succes van het tekenen afhangt van het aantal oneven knopen. Dat kunnen we ook zonder wiskundige kennis wel begrijpen. We kunnen de figuur, een *graaf* zeggen we, opvatten als een routekaart van een busonderneming waarin de busstations knopen worden genoemd. Het is dan de kunst om met een bus ergens te beginnen en alle stations aan te doen waarbij elke weg slechts eenmaal wordt bereden. Komt de chauffeur al rijdend bij een bepaald station aan, dan moet hij daar via een andere route weer vertrekken. Komt er bij dat station een even aantal wegen samen dan kan hij na een aankomst altijd weer vertrekken. Het aantal nog te berijden wegen vermindert telkens in paren van twee. Maar komt er ergens een oneven aantal samen dan blijft er ten slotte een enkele weg over die natuurlijk nog bereden moet worden. Dat gaat alleen maar goed aan het begin of aan het eind van de totale rit.

Voor het welslagen van een rit is het dus nodig dat er niet meer dan twee oneven knooppunten van wegen zijn. Zijn er inderdaad twee, zoals in figuur 8.1 (de punten aan de onderzijde), dan dient de rit in het ene punt te beginnen om dan ten slotte in het andere oneven knooppunt te eindigen. Zijn er alleen even knopen aanwezig, dan kunnen we het begin-



figuur 8.1 Een huisje als doorlopende lijn

figuur 8.2
De bruggen van Koningsbergen



punt vrij kiezen, maar wanneer het aantal knopen groter is dan twee zal het nooit lukken.

Leonhard Euler (1707-1783) was in zijn tijd een der grootste en meest produktieve wiskundigen. Zowel zuivere als toegepaste wiskunde hadden zijn aandacht. In zijn *Lettres à une princesse d'Allemand* toonde hij ook de wetenschap op vaardige wijze te kunnen populariseren. In zijn tijd stroomde in het toenmalige Koningsbergen de rivier de Pregel waarin zich tussen beide oevers een eiland bevond. De stadsdelen en het eiland waren onderling verbonden door zeven bruggen (figuur 8.2). Men vroeg zich toen af of het mogelijk was een dusdanige wandeling uit te voeren dat elke brug slechts eenmaal gepasseerd werd en dat men weer tot het punt van uitgang zou terugkeren. Euler liet zien dat dat niet mogelijk was. Hij simuleerde de wandeling door een graaf met knopen en trok uit de aanwezigheid van te veel oneven knopen de juise conclusie. En passant werd hij zo de schepper van de grafentheorie.

De schrijvende schildpad

Ouder geworden beschikken we over modern tekengereedschap en dat is tegenwoordig de computer, een machtig hulpmiddel dat veel mogelijk maakt maar dat in principe net zo dom is als zijn gebruiker. Onderwijsdeskundigen vinden dat kinderen maar vroeg aan de computer moeten wennen, en een van de betere methoden daarbij is het gebruiken van wat in de computerwetenschap *turtlegraphics* heet. "LOGO"

De kinderen zien dan een soort schrijvende schildpad zich over het beeldscherm bewegen die met eenvoudige toetsinstructies te besturen is:

beweeg en teken een recht lijntje
maak een kwartslag naar links
maak een kwartslag naar rechts

In deze eenvoudigste versie hebben we nog helemaal geen dure computer nodig, ruitjespapier en een pen of potlood zijn voldoende. Het tekenen van een schuine lijn geeft moeilijkheden, maar daar is al zigzaggend wel wat aan te doen.

De commando's voor de schildpad kunnen we aanduiden door symbolen als F, + en -. De schildpad kenmerken we door een drietal getallen x, y en φ. De getallen x en y geven de positie van de schildpad in rechthoekige coördinaten waarvan we de oorsprong vaak in het centrum van het beeldvlak kiezen. Op papier komt dat neer op tellen van ruitjes ten opzichte van een verticale of horizontale lijn door het centrum. Het derde getal geeft de richting waarin de schildpad zich beweegt. Werken we in graden dan correspondeert $\varphi = 0$ met horizontaal naar rechts, $\varphi = 90$ met verticaal omhoog enzovoort. De opdrachten + en - betekenen dan dat φ telkens met een bedrag van 90° vermeerderd of verminderd wordt. Eventuele grote waarden van de hoek kunnen we natuurlijk altijd reduceren tot een waarde tussen zeg 0 en 360°. Een hoek van bijvoorbeeld φ = 630° kunnen we opvatten als een hoek van 270° of zo men wil van -90°.

We spreken af dat we aan het begin de toestand (x,y,φ) van de schildpad aangeven als (0,0,0). De opdrachten + en - kunnen we wiskundig beschrijven als

$$+ \qquad \varphi \rightarrow \varphi + 90$$
$$- \qquad \varphi \rightarrow \varphi - 90$$

De opdracht F betekent dan het tekenen van een lijnstukje met een vaste lengte in de richting die de schildpad dan heeft. Die lengte moet van geval tot geval van tevoren worden vastgesteld. Op ruitjespapier zou men de lengte van een of van twee ruitjes kunnen nemen.

In het programma TURTLE is het een en ander nader uitgewerkt. Het programma bestaat eigenlijk uit een telkens te herhalen lus die reageert op een toetsaanslag. Met de toets F schrijft de schildpad een recht lijntje. Met de plus-toets wordt de richting veranderd in een kwartslag naar links (in positieve zin volgens de wiskundige conventie) en met de min-toets draait de schildpad een kwartslag naar rechts. Aldus kunnen we met een aantal toetsaanslagen de schildpad een uit horizontale en verticale lijntjes opgebouwde weg laten afleggen. Het programma laat zien dat met een X- of x-aanslag het einde is bereikt. Heel vaak worden in een programma de toetsen X, x, Q, q gebruikt om te eindigen (X = eXit, Q = Quit). De staplengte is aan het begin als 10 vastgesteld, maar dat getal kan natuurlijk veranderd worden. Gebruikers van een oudere Basic-versie kunnen hier weer kennis maken met de voordelen van een moderne programmeertaal, een betere lusstructuur, een handige meerkeuzetoets en geen regelnummers.

De taal van de schildpad

We hebben waarschijnlijk al gemerkt dat de schildpad best tot een reeds afgelegd stukje weg kan terugkeren. Bezwaarlijk is dat niet, maar misschien wel dat de schildpad almaar blijft schrijven. We breiden daarom

het repertoire wat uit door het symbool f te interpreteren als een verplaatsing, net als F maar dan zonder schrijven. Ook zou het prettig zijn de schildpad schuine lijntjes te kunnen laten trekken. We verruimen daartoe de betekenis van de symbolen + en - door ze te interpreteren als een richtingverandering in positieve of negatieve zin over een hoek α van nader te bepalen grootte. De complete lijst van schildpadinstructies is nu:

F	verplaatsing over de lengte h met tekenen
f	verplaatsing over de lengte h zonder tekenen
+	richtingverandering linksom over de hoek α
-	richtingverandering rechtsom over de hoek α

Om de richtingverandering van de schildpad in het programma weer te geven kunnen we twee wegen inslaan. De eenvoudigste methode is om de verplaatsing van de schildpad die zich in de richting φ bevindt, te beschrijven met de regels

$$x \rightarrow x + h\cos(\varphi)$$
$$y \rightarrow y + h\sin(\varphi)$$

en de hoekverandering simpel als

$$\varphi \rightarrow \varphi \pm \alpha$$

Dat werkt theoretisch voortreffelijk, maar in de computerpraktijk zijn er problemen. Bij een forse tekening moeten vele duizenden malen sinussen en cosinussen uitgerekend worden. Dat kost niet alleen tijd, maar is ook onnauwkeurig. Meestal trekken we ons niet al te veel aan van de nauwkeurigheid waarin onze computer werkt, maar in de techniek van de turtlegraphics kunnen kleine foutjes dusdanig accumuleren dat de tekening er slecht gaat uitzien.

Beter is het om gebruik te maken van de draaiingsformules van de coördinatenmeetkunde

$$p \rightarrow ap - bq$$
$$q \rightarrow bp + aq$$

die uitdrukken dat de vector (p,q) in positieve zin gedraaid wordt over de hoek α waarbij

$$a = \cos(\alpha) \qquad \text{en} \qquad b = \sin(\alpha)$$

De richting van de schildpad wordt dan nog steeds door de hoek φ gemarkeerd, maar in plaats van de hoek φ gebruiken we het tweetal (p,q), de cosinus en sinus van φ. Het grote voordeel is dan winst van snelheid en van precisie.

Om een beetje interessante figuur te krijgen moeten we heel veel kleine lijntjes tekenen. Het zou heel dom zijn wanneer we een dure computer zouden gebruiken om met de hand via het toetsenbord de schildpad zeg

enige honderden tekeninstructies te geven. Die tekenopdrachten vormen een lang woord van vele letters en dat woord zou eigenlijk veel beter met behulp van de computer samengesteld moeten kunnen worden. Voor een aantal interessante meetkundige figuren waarin veel regelmaat en symmetrie verscholen zit blijkt dat op verrassende wijze goed mogelijk te zijn.

Hoe de schildpad een fractal tekent

Als voorbeeld nemen we de lijn van Von Koch, een klassiek voorbeeld van een fractale lijn. In figuur 8.3 zien we de eerste twee stadia.

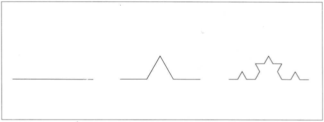
Een recht lijnstuk wordt kennelijk vervangen door een gebroken lijn die uit vier gelijke stukjes bestaat. In schildpadtaal betekent dat

$$F \rightarrow F + F - - F + F$$

Het symbool + betekent nu een richtingverandering over een hoek van 60° in positieve zin terwijl - hetzelfde in negatieve zin betekent. Met de meetkundige constructie van de Von Koch-lijn gaan we nog een stapje verder en we vragen ons af hoe het nieuwe woord van commando's eruit ziet. Meetkundig wordt elk recht lijntje, waarvan er nu vier voorhanden zijn, vervangen door de gebroken lijn van vier stukjes uit figuur 8.3. Voor de schildpad betekent dat dat in zijn aanvankelijke instructiewoord elk van de vier letters F vervangen moet worden door het woord F+F- -F+F. We noemen dat een *produktieregel* en die hebben we zojuist opgeschreven. Het nieuwe woord wordt dus

$$(F+F--F+F) + (F+F--F+F) -- (F+F--F+F) + (F+F--F+F)$$

De haakjes zijn er ter wille van de duidelijkheid aan toegevoegd maar kunnen natuurlijk worden weggelaten. Het zojuist opgeschreven woord bevat alles wat de schildpad nodig heeft om een volgende fase van de Von Koch-lijn te kunnen tekenen. Bovendien weten we nu ook precies hoe we verder kunnen gaan. Weer moet elke letter F volgens de produktieregel door een woord van acht symbolen worden vervangen. Op deze wijze kunnen we verder gaan met steeds langere woorden. Het is een aardige



figuur 8.3 Het begin van de lijn van Von Koch

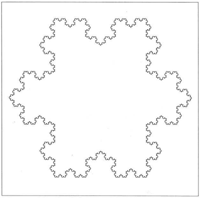
opgave voor de lezer om uit te rekenen hoe groot de volgende woorden worden. Maar het is inmiddels wel duidelijk dat we de computer hard nodig hebben. We gebruiken daartoe het programma TURTLEK. Dat programma bestaat uit twee delen. In het eerste gedeelte wordt voor de gewenste orde het instructiewoord gevormd, te beginnen met de enkele letter F, het zogenaamde *axioma*.

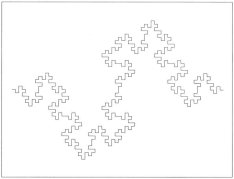
Dat zelfde programma, eenmaal beschikbaar en liefst ook begrepen, kunnen we dan voor andere doeleinden gebruiken. De enige verandering bestaat in een andere keuze van axioma en produktieregel. Ook moet de waarde α van de rotatiehoek van tevoren worden vastgelegd. Om te zorgen dat de figuur goed in het beeldscherm past is enige schaling noodzakelijk, hetgeen het beste proefondervindelijk kan geschieden. Het is dan verstandig om te beginnen met een lage orde als bijvoorbeeld 3. Door de staplengte *h* voldoende klein te nemen en de beginpositie aan te passen kunnen we de figuur dan mooi op het scherm krijgen. Met een dergelijke wijziging kunnen we het programma dan een zogenaamde sneeuwvlok laten tekenen, een samenstel van zes Von Koch-lijntjes. Eigenlijk behoeft alleen het axioma gewijzigd te worden. De gegevens vindt men hieronder, gelijk met nog een paar aardige voorbeelden. We vestigen er nog even de aandacht op dat het axioma niets anders is dan een beschrijving van de beginfiguur. Dat kan een enkel lijnstuk (F) zijn, een vierkant (F+F+F+F, maar ook F-F-F-F in een tegengestelde rondgang), enzovoort.

Voorbeelden

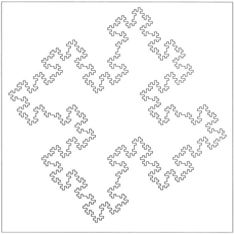
In de figuren 8.4 tot en met 8.9 laten we wat voorbeelden van turtlegraphics zien. De techniek van turtlegraphics is in principe heel eenvoudig. Door steeds maar een zelfde elementair recept te volgen wordt een lang woord, een string, gevormd dat alle instructies bevat die nodig zijn om de schildpad de verlangde figuur te laten vormen. In de praktijk

figuur 8.4 De sneeuwvlok van Von Koch
Hoek 60°
Axioma F+F+F+F+F+F
Regel F+F--F+F
Orde 4.

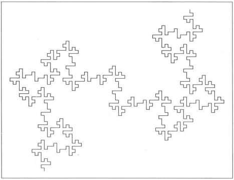




figuur 8.5
De lijn van Minkowski
Hoek 90°
Axioma F
Regel F-F+F+FF-F-F+F
Orde 3.

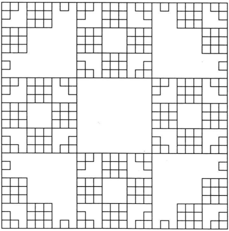


figuur 8.6
Vierkant Minkowski-eiland
Hoek 90°
Axioma F+F+F+F
Regel F-F+F+FF-F-F+F
Orde 3.

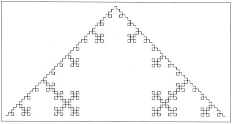


figuur 8.7
Variant van de Minkowski-lijn
Hoek 90°
Axioma +F
Regel F+F-F-FFF+F+F-F
Orde 3.

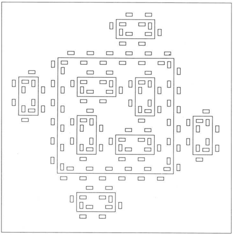
figuur 8.8
Vierkant van Sierpinski
Hoek 90°
Axioma F+F+F+F
Regel FF+F+F+F+FF
Orde 3.



figuur 8.9 Kwadratische variant van de Von Koch-lijn
Hoek 90°
Axioma F
Regel F+F-F-F+F
Orde 4.



figuur 8.10
De archipel van Mandelbrot
Hoek 90°
Axioma F+F+F+F
Regel1
F→F-f+FF-F-FF-Ff-FF+f-FF+F+FF +Ff+FFF
regel2 f→ffffff



worden aan de lengte van de string beperkingen gesteld afhankelijk van het type computer en van het besturingssysteem. Menig computergebruiker kent mededelingen als 'out of string space'. Gelukkig kan dat inconveniënt vermeden worden door gebruik te maken van de techniek van backtracking. In het volgende hoofdstuk gaan we er dieper op in. Het komt erop neer dat het vormen van de instructiestring op een zodanige wijze geschiedt dat al meteen met het tekenwerk begonnen kan worden zodra een stukje van het eindresultaat gereed is.

Bij het volgende voorbeeld (programma TURTLEA) – ontleend aan het bekende boek van Mandelbrot – groeit het woord al zo snel aan dat een derde fase al moeilijkheden kan geven. Hier onderscheiden we twee verschillende produktieregels, programmatechnisch gemakkelijk in een extra CASE-opdracht onder te brengen.

In figuur 8.10 laten we zien hoe dat eilandenrijk er na drie stappen uitziet.

De lijn van Lévy en de drakenlijn

In de vele publikaties over fractals komt de drakenlijn uitvoerig aan de orde. Het ligt dus voor de hand het tekenen ervan eens aan de schildpad voor te leggen. Daarbij doet zich een kleine complicatie voor, het is namelijk nodig om twee soorten lijnstukken te gaan onderscheiden. Een lijnstuk waarbij iets aan de linkerzijde, en een waarbij iets aan de rechterzijde moet gebeuren, tenminste wanneer we de conventie aanhouden dat de door de schildpad gemaakte gebroken lijn uit een aaneenschakeling van vectoren, kop aan staart geordend, bestaat.

Duiden we een gericht lijnstuk, een vector dus, waarbij iets aan de linkerkant moet gebeuren op de gebruikelijke wijze aan met het symbool F, en kiezen we een andere letter G bijvoorbeeld voor een overeenkomstige vector waarbij iets aan de rechterzijde moet gebeuren, dan moeten we twee verschillende produktieregels onderscheiden.

Voor de gewone winkelhaak geïllustreerd in figuur 8.11a lezen we de volgende produktieregel af:

+ F - - F +

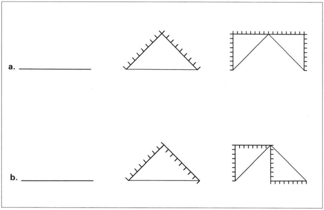
waarbij + de betekenis heeft van een draai over 45°. Wanneer we in het overeenkomstige computerprogramma de orde langzaam laten oplopen, zeg van 1 tot aan 10 (en wanneer we tevens de elementaire staplengte h aanpassen), dan krijgen we een figuur, de lijn van Lévy, die zich in alle gevallen van links naar rechts over het scherm uitstrekt. Dat komt omdat in de produktieregel de totale draaihoek, de som van alle + en - symbolen, nul is. Het bezwaar van deze methode is evenwel dat de gevormde string snel aangroeit en voorbij de orde 10 niet meer in het stringgeheugen past. We hadden al gezegd dat dit bezwaar met een betere programmeertechniek ondervangen kan worden, maar het kan hier ook eenvoudiger. De produktieregel

F - F (of F + F)

geeft precies hetzelfde resultaat, zij het dat de lijn van Lévy in zijn geheel om het beginpunt geroteerd is. Is p de gebruikte orde, dan is de totale draaihoek gelijk aan $p \star 45^\circ$. Het grote voordeel is dat de string veel minder snel aangroeit zodat $p = 12$ normaal gesproken gemakkelijk haalbaar is.

De tweede vorm van de winkelhaak, geïllustreerd in figuur 8.11b, de vorm die ten grondslag ligt aan de drakenlijn, geeft aanleiding tot twee produktieregels

F → + F - - G +
G → - F + + G -



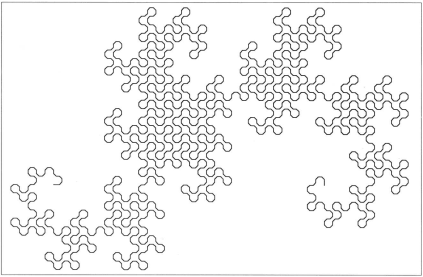
figuur 8.11 Twee manieren om een winkelhaak te maken

Uiteraard beginnen we in dit en in het vorige geval met een enkel lijnstuk. Dat betekent dus dat in beide gevallen het axioma 'F' is. Hierbij geldt een overeenkomstig commentaar als voor de lijn van Lévy. De totale draaihoek is nul zodat de drakenlijn netjes van links naar rechts loopt. Maar ook hier loopt de grootte van de symbolenstring snel uit de hand. Dat kan ook weer worden ondervangen met de vereenvoudigde versie

F → F - G
G → F + G

waarbij voor + en - een rotatie over 90° geldt.

Het met deze regels werkende programma TURTLED kan dan snel een resultaat als figuur 8.12, de tiende groeifase, op het scherm toveren.



figuur 8.12 De drakenlijn in zijn tiende groeifase

9 Lijnen die een vlak vullen

Wanneer we in onze huiskamer ontspannen, achterover geleund in een gemakkelijke stoel, naar televisiebeelden kijken beseffen we niet of nauwelijks wat er allemaal komt kijken voor de overdracht van een enkel beeld. Een beeld is begonnen als een tweedimensionaal plaatje, wordt gecodeerd als een eendimensionaal signaal dat via antenne of kabel in onze apparatuur wordt opgevangen, en daarin wordt op een omgekeerde wijze het oorspronkelijke beeld weer samengesteld. In abstracte termen gaat het hier om een tamelijk fundamenteel wiskundig probleem. Hoe kunnen we de punten van een rechthoek of een vierkant afbeelden op de punten van een rechte lijn? Gemakshalve nemen we aan dat we te maken hebben met een vierkant.

De eenvoudigste oplossing, en ook de oplossing die in de telecommunicatie is gevolgd, is om het beeld te verdelen in een aantal horizontale lijnen, en om die lijnen van boven naar beneden af te tasten, en elke lijn van links naar rechts te bewerken. Het doet denken aan een methode om de velden van een schaakbord op één lange rij van 64 te plaatsen, eerst de achtste rij van links naar rechts (de velden a8 tot en met h8), dan de zevende rij (met de velden a7 tot en met h7) enzovoort tot aan de eerste rij. Maar daar zit wel een probleem aan vast dat zich op onaangename wijze openbaart wanneer er een storing is. In de huiskamer zien we dan op onze beeldbuis horizontale strepen lopen die het gewenste beeld kunnen bederven. Een wetenschapper vraagt zich allicht af of het niet beter zou kunnen. Het bezwaar bij de zojuist beschreven methode is dat beeldpunten die in het vierkant dicht bij elkaar liggen op de lijn wel eens ver uit elkaar terecht kunnen komen. In het voorbeeld van het schaakbord blijven c2 en d2 wel naast elkaar, maar c2 en c3 zijn uiteengedreven. We vragen ons dus af of er niet een methode bestaat waarbij het dicht bij elkaar zijn niet wordt verstoord.

De lijn van Peano

Welnu, die methode bestaat en is reeds in 1885 door de Italiaanse wiskundige Giuseppe Peano bedacht. Peano liet zien dat het mogelijk was een vierkant op continue wijze te vullen met een enkele doorlopende lijn. Het lijkt een vreemde zaak, en de wiskundigen hadden er in die tijd veel moeite mee. Duidelijk is wel dat aan de definitie van een lijn heel wat meer vastzit dan men oppervlakkig zou denken. Nu, een honderd jaar later, is men vertrouwd geraakt met de wonderlijke wereld van fractals en ook een vlakvullende lijn is niets ongewoons meer. Aan het eind van het vorige hoofdstuk hebben we er al een voorbeeld van gezien, de bekende

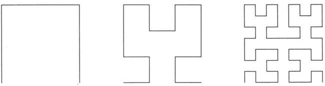
drakenlijn. De door Peano gevonden lijn is natuurlijk een wiskundige idealisatie, maar een benadering van een dergelijke lijn is in de praktijk heel goed bruikbaar. Na Peano hebben andere wiskundigen soortgelijke constructies bedacht, met name David Hilbert, een van de belangrijkste wiskundigen bij de aanvang van de twintigste eeuw. Gecombineerd met de moderne computertechniek kan het gebruik van een Peano-lijn leiden tot een doeltreffende methode om op systematische wijze een fysisch oppervlak te 'scannen'.

Afgezien van de theoretische en technische merites van Peano-achtige lijnen zijn ze een dankbaar object voor de computer. We zullen dan ook laten zien hoe de in het vorige hoofdstuk geïntroduceerde schildpad op grond van een paar eenvoudige regels een lijn van Peano kan beschrijven. Als voorbeeld nemen we de versie van David Hilbert (figuur 9.1).

De schrijvende schildpad kan met behulp van het axioma X en de twee produktieregels

X → - YF + XFX + FY -
Y → + XF - YFY - FX +

het tekenwerk op een robotachtige wijze tot uitvoering brengen. Het symbool F is het tekenen van een lijnstuk, + betekent een kwartdraai naar links (of rechts) en - een kwartdraai naar rechts (of links). De symbolen X en Y zijn alleen van belang voor de vorming van de string en worden door de schildpad bij het tekenen niet gebruikt. De lezer doet er verstandig aan om zelf, gewapend met potlood en ruitjespapier, een klein stukje mee te tekenen. Men begint met het potlood ergens in een roosterpunt te plaatsen. Het axioma X, de nulde fase, brengt de schildpad nog niet in beweging. De eerste fase levert volgens de eerste produktieregel de string -YF+XFX+FY-. Staat de schildpad bij de startpositie naar rechts gericht, dan wordt een soort U getekend, een stapje naar beneden, naar rechts en weer omhoog. Ten slotte heeft de schildpad weer de oorspronkelijke oriëntatie. In de tweede fase worden de symbolen X en Y volgens de produktieregels elk vervangen door een deelstring van elf symbolen. De totale string bestaat nu uit 51 symbolen. Daarin zitten vijftien symbolen F, zodat het meetkundige equivalent een gebroken lijn is van vijftien stukjes. De symbolen X en Y hebben geen directe meetkundige betekenis, maar men zou ze kunnen interpreteren als posities op de gevormde kronkellijn waar in de volgende fase nieuwe 'u-tjes' gevormd moeten worden.



figuur 9.1 Drie stadia in de vorming van een Peano-lijn

We merken nog even op dat de tweede produktieregel in zekere zin de omgekeerde of symmetrische is van de eerste regel. Uit de illustratie volgt dat die symmetrie ook van meetkundige aard is. In het programma PEANO is dit alles op eenvoudige wijze verwerkt.

Backtracking

Wie dit of een soortgelijk programma wil uitvoeren in een hogere orde komt vermoedelijk in botsing met de beperkingen die MS-DOS stelt aan het stringgeheugen. Daar kan op verschillende wijzen wat aan gedaan worden. Bij sommige programmeertalen kan men met enige handigheid wel op een zuiniger wijze met strings omgaan. Een andere en wellicht betere manier is die van *backtracking*, waarbij men niet eerst het – vaak heel lange – woord van de laatste fase opbouwt alvorens te gaan tekenen. Het vormen van de eindstring kan men opvatten als het construeren van een abstracte boom waarbij een symbool van het axioma de stam is, en waarbij de takken zich volgens de produktieregels splitsen in volgende takken. De symbolen van de uiteindelijke string zitten dan als knopjes aan de uiteinden van de takken van de hoogste orde. Bij het doorlopen van de boom gaan we telkens weer naar zo'n knopje toe, waarbij we onderweg bij elke splitsing de nog niet gebruikte takken op een stapel bewaren voor later. Voor wie dat mechanisme kent zal het nu wel duidelijk zijn, maar ter wille van de minder ervarenen leggen we aan de hand van een eenvoudig voorbeeld het een en ander uitvoeriger uit.

Als axioma kiezen we het enkele symbool F en we werken met de produktieregel

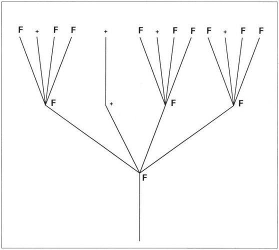
$$F \rightarrow F + FF$$

In drie stappen krijgen we dan

F   F+FF   F+FF+F+FFF+FF
F+FF+F+FFF+FF+F+FF+F+FFF+FFF+FF+F+FFF+FF

In figuur 9.2 hebben we de corresponderende boomstructuur getekend waarbij we ter wille van de duidelijkheid tot aan de tweede stap zijn gegaan. Algemeen gesproken zijn er zoveel soorten takken als er symbolen zijn. De splitsing van een tak is bepaald door de bijpassende produktieregel. Dat betekent hier dat een F-tak zich splitst in drie F-takken en een +-tak, en dat een +-tak zonder splitsing overgaat in een volgende +-tak. Bij dit alles is de volgorde natuurlijk erg belangrijk.

De boom wordt doorlopen van beneden naar boven, te beginnen met de F van de stam, het axioma. Bij de splitsing zetten we eerst alle vier splitsingen even op een stapel in de volgorde FF+F, dus achterstevoren de produktieregel volgend. Het bovenste symbool halen we van de stapel en daarmee gaan we verder. Dat is dus een F en ook daarna leggen we de symbolen van de vier splitsingen weer omgekeerd op de stapel. De bovenste halen we eraf en nu zijn we aan de kruin van de boom, tenminste wanneer we ons als in dit plaatje tot de tweede fase beperken. De F rechts



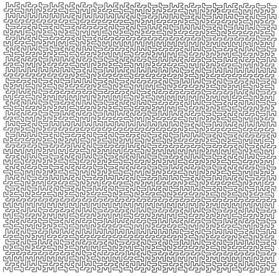
figuur 9.2
Het schema van backtracking

bovenaan in de kruin bieden we aan ter verwerking door het tekenprogramma. Vervolgens pakken we het bovenste element van de stapel, die nu zes symbolen bevat. Dat is een + die ook aan de kruin zit evenals de volgende twee symbolen F. Ze worden achtereenvolgens verwerkt door het tekenprogramma en daarna gaan we verder met de stapel, die nog drie elementen bevat uit fase 1. De bovenste is een + en die geeft ook een + in de tweede fase, zodat het grafische programma weer een actie kan uitvoeren. Daarna is de bovenste een F en die levert volgens de produktieregel vier takken. Met de bovenste gaan we verder en de overige drie leggen we even op de stapel tot ze aan de beurt kunnen komen. Er gebeurt nu verder alleen maar een herhaling van bekende zaken totdat de stapel leeg is en het proces eindigt. Eigenlijk moeten we bij deze manipulaties naast de symbolenstapel ook een ordestapel bijhouden, zodat we bij elk element dat aan de beurt komt tevens de bijbehorende fase kunnen weten. Die wetenschap kunnen we gebruiken om te bepalen wanneer we de kruin van de boom bereikt hebben, uiteraard wanneer we van tevoren hebben afgesproken tot welke eindfase we willen gaan.

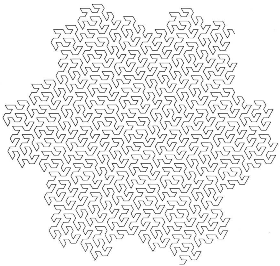
Deze methode is gebruikt in het programma PEANOB met behulp waarvan men een hogere fase als geïllustreerd in figuur 9.3 op het scherm kan krijgen.

Het is nu een vrij eenvoudige zaak geworden soortgelijke Peano-lijnen te genereren. Het enige is een andere keuze van axioma en produktieregel(s) en een corresponderende meetkundige interpretatie. We laten het aan de lezer over om Peano-krommen met de volgende produktieregels te vormen:

Axioma F+XF+F+XF
Produktieregel X → XF-F+F-XF+F+XF-F+F-X



figuur 9.3 Een Peano-lijn volgens Hilbert



figuur 9.4 Een hexagonale lijn van Peano volgens Gosper

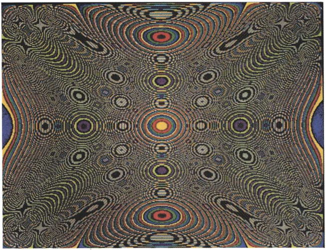
Axioma X
Regel1 X → XFYFX+F+YFXFY-F-XFYFX
Regel2 Y → YFXFY-F-XFYFX+F+YFXFY

Tot slot geven we in figuur 9.4 een afbeelding van de zogenaamde lijn van Gosper, een van de bekende plaatjes uit Mandelbrots boek *The Fractal Geometry of Nature*. De produktieregels zijn:

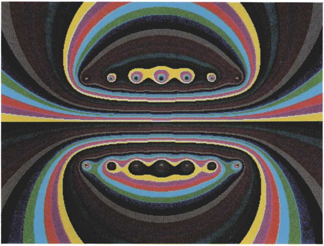
Axioma XF
Fig 9.4 Regel1 X → X+YF++YF-FX- -FXFX-YF+
Regel2 Y → -FX+YFYF++YF+FX- -FX-Y

Het ziet er op het eerste gezicht nogal geducht uit, maar net als bij het bovenstaande voorbeeld b is regel2 niets anders dan een gespiegelde versie van regel1, een symmetrie die in de illustratie zijn meetkundige vertaling vindt. Interpreteren we + en - als draaiingen over ± 60°, dan geeft de combinatie van het axioma met regel1 (met negeren van X en Y) de hoofdvorm van de kronkellijn.

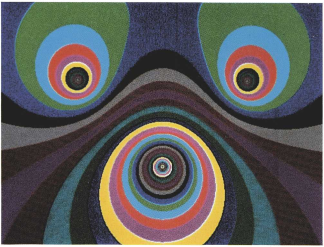
Figuur 3.3 De hoogtekaart van een bergpas in kleur



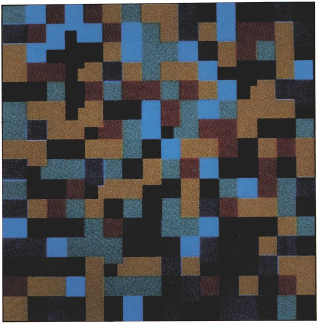
Figuur 3.8 Een kleurig patroon met moiré



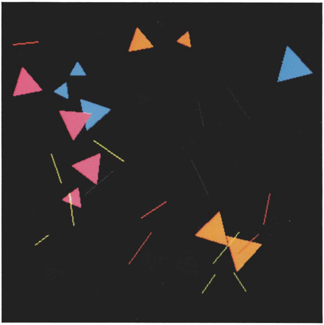
Figuur 5.5 Het veld van een condensator



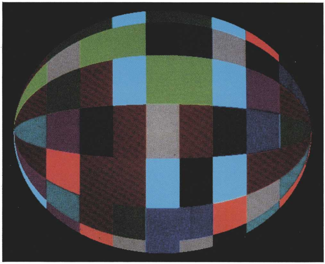
Figuur 5.6 Het veld bepaald door twee bronnen en een put



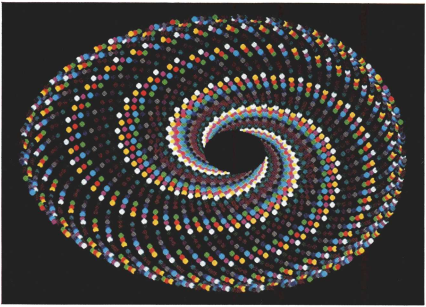
Figuur 7.2 Een toevallig geblokt dessin



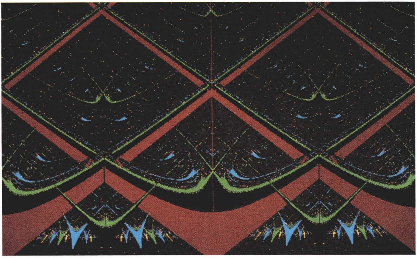
Figuur 7.3 Een voorstelling op de wijze van Klee



Figuur 7.5 De projectie van een geblokt ei



Figuur 7.6 Een ei bepaald door een dynamisch systeem



Figuur 15.9 *c,a*-Diagram van fasekoppeling

Figuur 16.7 Het Lyapunov-diagram van het Markus-model aabab

10 Van woord tot plant en boom

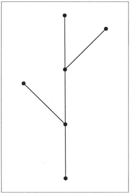
De in de hoofdstukken 8 en 9 beschreven methode om woorden te vormen die een grafische interpretatie toelaten is in 1968 door de Utrechtse bioloog Aristides Lindenmayer uitgedacht. Het was zijn streven om biologische groei op een zo abstract mogelijke wijze te simuleren. Hij bediende zich daartoe van een soort primitieve taal met een beperkt *alfabet* van symbolen of letters waarmee een woord gevormd wordt met behulp van zogenaamde produktieregels, waarbij een enkel symbool vervangen werd door een woord. Het allereerste woord, men moet tenslotte ergens beginnen, noemde hij het axioma. Zijn werk trok de aandacht van informatica-deskundigen die zich met zogenaamde abstracte computertalen bezighielden. Kortom, het luidde een nieuwe ontwikkeling in, een nieuw vakgebied met meer dan duizend wetenschappelijke bijdragen. Aanvankelijk paste Lindenmayer zijn zogenaamde *L-systemen* alleen toe op eencellige organismen, maar weldra konden er ook planten en bomen mee gesimuleerd worden.

In hoofdstuk 8 hebben we gezien hoe effectief zo'n L-systeem voor de vorming van fractale lijnen kan zijn. In hoofdstuk 9 hebben we daaraan nog een vervolg gegeven met de constructie van Peano-lijnen. Hier gaan we op de ingeslagen weg verder, en breiden we de schildpadtaal zodanig uit dat we bijvoorbeeld de groei van een zich vertakkende boom kunnen weergeven. Tot dusverre gebruikten we in principe slechts de symbolen F, f, + en -. Daar kan dan nog G als alternatief voor F bijkomen, en een enkele maal hebben we ook dummy-symbolen als X en Y gebruikt. Aan ons repertoire voegen we nu de symbolen [en] toe. Het symbool [(haakje openen) betekent voor de schildpad dat hij zich in een vertakkingspunt bevindt en dat hij daar later naar moet terugkeren om de volgende daar aangehechte tak(ken) te kunnen vormen. Dat betekent natuurlijk wel dat de schildpad met een soort geheugen begiftigd moet zijn om die plaats voor later gebruik te kunnen onthouden. Zodra de schildpad bij het lezen en uitvoeren van de opdrachten een sluithaak] tegenkomt dient hij te weten dat het einde van een vertakking bereikt is en dat hij moet terugkeren tot het vertakkingspunt waar hij het laatste is geweest.

In figuur 10.1 laten we zien hoe de schildpad een uit de verschillende symbolen gevormd woord interpreteert. De te gebruiken hoek kiezen we daarbij als 45°:

F[-F][+F[+F]F]F[-F][+F]

De vraag rijst hoe we de schildpad van een eenvoudig doch efficiënt geheugen kunnen voorzien. Zoals we gezien hebben is de positie van de schildpad op elk moment vastgelegd door twee plaatscoördinaten x, y en

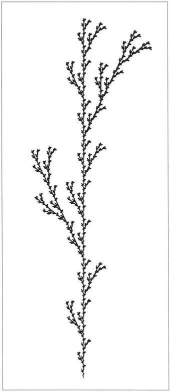


figuur 10.1 Hoe de schildpad van een woord een takje maakt

een hoek φ, die de looprichting aangeeft. Zodra de schildpad een vertakking, een knoop van waaruit verschillende voortzettingen mogelijk zijn, tegenkomt dient hij dus die drie getallen te noteren. Hij doet dat op een soort kaartje – denk maar aan een speelkaart. Telkens wanneer de schildpad een nieuwe splitsing tegenkomt schrijft hij de gegevens op een kaartje en legt hij het nieuwe kaartje bovenop het oude. Op die wijze kan een stapeltje van kaarten ontstaan, iets wat in computerjargon als een *stack* beschreven wordt en wat wij gewoon een *stapel* noemen. Zodra de schildpad met het symbool] het einde van een tak bereikt behoeft hij slechts naar de bovenste kaart van de stapel te kijken om te weten waar hij moet terugkeren en welke richting hij dan moet inslaan. De kaart is dan niet meer nodig en wordt verwijderd. We zien aldus in onze verbeelding een stapel kaartjes van wisselende samenstelling ontstaan. Het enige wat er gebeurt is een kaart op de stapel leggen of een kaart van de stapel weghalen. De stapeltechniek wordt vooral gebruikt bij de methode van backtracking, die in vorige hoofdstukken een paar maal ter sprake is gekomen. In de techniek van het programmeren wordt een stapel veelvuldig gebruikt. Het hart van menige computer, de centrale microprocessor zoals die van het 80 ⋆ 86 type, bevat dan ook standaard een zogenaamd stapelregister. Maar wanneer wij een programma willen schrijven voor het genereren van een of ander struikje dienen we zelf in het programma een stapel in te voeren. Moeilijk is dat niet, en het programma LIND1 laat zien hoe dat in zijn werk gaat. Eerst zetten we de verschillende symbolen van de schildpadtaal nog even op een rijtje. Met de positie van de schildpad bedoelen we daarbij de twee plaatscoördinaten x, y en de bewegingsrichting φ.

F	Ga een stap verder en teken
f	Ga een stap verder zonder te tekenen
+	Draai over een hoek α naar links (of rechts)
-	Draai over een hoek α naar rechts (of links)
[	Plaats de positie op de stapel
]	Haal de positie van de stapel en breng de schildpad in die nieuwe positie zonder te tekenen

Het vrij eenvoudige programma LIND1 stelt ons in staat het plantje van figuur 10.2 te vormen. Het bevat – vergeleken met overeenkomstige programma's uit de hoofdstukken 6 en 8 – eigenlijk alleen het stapelgebruik als nieuwigheid. Vooraf reserveren we in de arrays u(),v(),w() enige plaatsruimte voor het opslaan van gegevens. Het aantal te gebruiken plaatsen is vooraf onbekend, maar met enig proberen lijkt zestien wel een goede keus te zijn. In de praktijk kunnen we natuurlijk voor de veiligheid meer ruimte reserveren om achteraf te bekijken of het zuiniger kan. Het niveau van de stapel, het rangnummer van de kaart met gegevens, duiden we aan met de index m. De constructie van het instructiewoord in een aantal fases is een min of meer abstract formalisme waarbij de meetkundige betekenis van de gebruikte symbolen er niet toe doet. Die komt pas bij het grafische gedeelte. Het formalisme van een kaart op de stapel plaatsen of daarvan verwijderen ziet men weerspiegeld in de laatste twee selectieregels CASE [en CASE]. Het resultaat van het programma ziet men in figuur 10.2.



figuur 10.2 Een plant met de regels
Axioma F
F → F[+F]F[-F]F
$\alpha = \pi/7$

Op analoge wijze kunnen planten of struiken worden gevormd met de modellen uit het volgende lijstje. Ter wille van het overzicht is het zojuist besproken model eraan toegevoegd.

1. Axioma: F (figuur 10.2)
   Regel : F → F[+F]F[-F]F
   met $\alpha = \pi/7$

2. Axioma: F (figuur 10.3)
   Regel : F → FF+[+F-F-F]-[-F+F+F]
   met $\alpha = \pi/6$

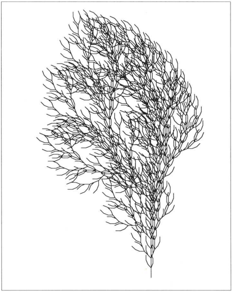
3. Axioma: X (figuur 10.4)
   Regel1: X → F[+X]F[-X]+X
   Regel2: F → FF
   met $\alpha = \pi/9$

4. Axioma: Y (figuur 10.5)
   Regel1: Y → YFX[+Y][-Y]
   Regel2: X → X[-FFF][+FFF]FX
   met $\alpha = \pi/7$

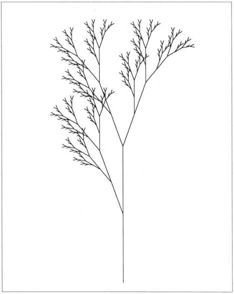
5. Axioma: X (figuur 10.6)
   Regel1: X → F-[[X]+X]+F[+FX]-X
   Regel2: F → FF
   met $\alpha = \pi/8$

Zoals we in de vorige hoofdstukken hebben opgemerkt krijgen we problemen met de computer wanneer we de planten in een hogere orde willen laten groeien. De string waaruit de opbouw van het groeisel bij de grafische bewerking moet worden afgelezen kan te lang worden voor het beschikbare stringgeheugen. Maar ook nu weer kan het veel handiger door gebruik te maken van de techniek van backtracking. We krijgen in het resulterende programma nu twee soorten stapels. De eerste stapel dient alleen voor de opbouw van de string in de hoogste groeifase, de tweede stapel heeft alleen betrekking op de grafische bewerking om de boekhouding van de vertakkingen te verzorgen.

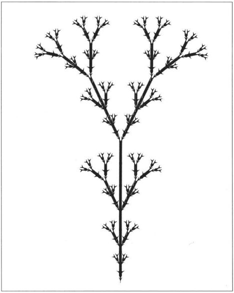
Het programma LIND2 is aldus een versie van het programma LIND1 waarbij nu backtracking is toegepast. Het heeft in de in het boek opgenomen vorm betrekking op het vijfde model, maar het kan met verandering van axioma en regels en met aanpassing van de beeldbegrenzing ook voor de andere modellen worden gebruikt.



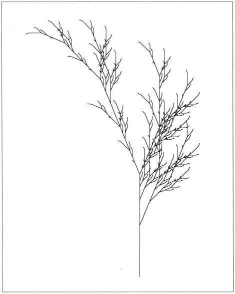
figuur 10.3 Een plant met de regels
Axioma F
F→FF+[+F-F-F]-[-F+F+F]
$\alpha = \pi/6$



figuur 10.4 Een plant met de regels
Axioma X
X → F[+X]F[-X]+X F → FF
$\alpha = \pi/9$



figuur 10.5 Een plant met de regels
Axioma Y
Y → YFX[+Y][-Y]
X → X[-FFF][+FFF]FX
$\alpha = \pi/7$



figuur 10.6 Een plant met de regels
Axioma X
X → F-[[X]+X]+F[+FX]-X
F → F
$\alpha = \pi/8$

11 Kartografie, het bedrog van landkaarten

De Nederlanden hebben een grote traditie in het maken van landkaarten. We hoeven slechts even de namen van Mercator, Ortelius en Blaeu te memoreren om daarvan overtuigd te zijn. Sindsdien heeft het handwerk van het maken van kaarten zich verder geperfectioneerd, en nu is dan ten slotte de computer ingeschakeld. Bij het maken van een bruikbare kaart als een afbeelding van een bolvormig oppervlak op een vlak vel papier komt heel wat kijken. Dat is allereerst een wiskundig probleem. Wiskundigen verzekeren ons al meteen dat het onmogelijk is een bol op een plat vlak af te beelden zonder dat er vervormingen optreden. We moeten dus concessies doen, en het hangt er maar helemaal vanaf voor welk doel we een landkaart of een zeekaart wensen te gebruiken. Het zou bijvoorbeeld prettig zijn wanneer op een kaart gebieden van gelijke grootte, dus met dezelfde oppervlakte, ook in werkelijkheid even groot zouden zijn. Een kaart die hieraan voldoet pleegt men vlaktrouw of *equivalent* te noemen. Wil men de kaart gebruiken om te navigeren dan zijn hoeken erg belangrijk. Een hoek tussen twee richtingen die op het aardoppervlak zeg 60° bedraagt moet op de kaart ook even groot zijn. Een dergelijke kaart noemt men hoektrouw of *conform*. Een bekende projectiemethode als die van Mercator is bijvoorbeeld wel conform maar niet equivalent. We weten uit eigen ervaring wel dat gebieden in de buurt van de poolstreken, Groenland bijvoorbeeld, er op kaarten die met deze methode zijn vervaardigd veel te groot uitzien.

In dit hoofdstuk willen we proberen enig inzicht te verwerven in de diverse projectiemethoden aan de hand van een paar computerexperimenten. Maar voordat we daartoe overgaan is het goed even in een atlas te kijken en ons te bezinnen op de problematiek van het maken van kaarten. Duidelijk is dat een afbeelding van een klein stukje van het aardoppervlak niet zoveel problemen geeft. Hoe kleiner het stukje des te geringer de afwijking tussen bolvorm en een plat vlak. Het ligt dan voor de hand een orthogonale projectiemethode te gebruiken waarbij het vlak het raakvlak is aan de aardbol in het centrum van het af te beelden stukje. Een vierkant stuk land ziet er op de kaart dan ook als een vierkant uit. De moeilijkheden komen pas wanneer we een groter stuk van het aardoppervlak willen afbeelden. Dat is in de eerste plaats een zaak van landmeting, van het bepalen van afstanden en van hoeken. Kijken we eens in de atlas naar een gebied op een hogere breedtegraad – ergens in Canada bijvoorbeeld – dan zijn we geneigd een gebied dat afgepaald is door vaste lengtegraden α en vaste breedtegraden β te beschouwen als een rechthoek. Inderdaad ziet het er op de gebruikelijke kaarten als een rechthoek uit, maar daar klopt in werkelijkheid helemaal niets van. Lengtemeting geschiedt op een boloppervlak, het aardoppervlak dus, langs grote cirkels, cirkels waarvan

het vlak door het middelpunt van de bol gaat. Grote cirkels zijn op aarde wat rechte lijnen op een plat vlak zijn, de kortste verbindingslijnen tussen twee punten, zogenaamde geodetische lijnen. Een vliegtuig volgt bij een transcontinentale vlucht zoveel mogelijk een grote cirkel. Meridianen zijn grote cirkels maar breedtecirkels zijn dat niet. Naarmate we op hogere breedtegraden komen zijn de afwijkingen tussen breedtecirkel en grote cirkel des te groter.

De kaarten in een atlas kunnen ons behoorlijk voor de gek houden wat betreft de werkelijke vorm en grootte van een of ander gebied, en dat geldt vooral voor kaarten waarop hele continenten te zien zijn. De computer kan ons hulp bieden om inzicht te krijgen in het soort bedrog dat de kaartenmakers met ons plegen. Daarmee is natuurlijk niet gezegd dat wij opzettelijk om de tuin worden geleid, maar is wel bedoeld te zeggen dat we met de interpretatie van geografische gegevens voorzichtig moeten zijn, en dat de computer ons daarbij als een goede gids kan leiden. Met de computer zijn we in staat van een willekeurig stukje aardoppervlak in een door ons zelf gekozen projectiemethode een detailkaart te maken, en we kunnen daarbij de resultaten van de verschillende kaartmethoden met elkaar vergelijken. Om niet in aardrijkskundige details te vervallen gaan we nu wat meer wiskundig te werk. We vragen ons bijvoorbeeld af hoe een vierkant stuk land, geplaatst op een willekeurige breedtegraad, er op een kaart zou uitzien.

Bij het maken van een kaart beperken we ons tot de orthogonale projectie van een bol op een plat vlak, het vlak van tekening dat samenvalt met het beeldscherm van de computer. Die bol, een model van de aarde, laten we draaien, en we vragen ons af wat we bij projectie op het beeldscherm te zien krijgen wanneer we een door geografische coördinaten gegeven eiland laten meedraaien. Daarbij zorgen we er ook voor dat we alleen het zichtbare gedeelte van de aardbol afbeelden.

De stilstaande aarde

Uiteraard moet alles in coördinaten worden uitgedrukt, en daartoe bedienen we ons van twee Cartesische coördinatenstelsels. Het eerste systeem, een x,y,z-stelsel, zit aan de aarde vast. Alles kiezen we zo eenvoudig mogelijk, de oorsprong in het middelpunt en de z-as langs de omwentelingsas. De straal van de aarde kiezen we als eenheid. Een willekeurig punt op de bol kunnen we beschrijven als

$$x = \cos\alpha\cos\beta, \qquad y = \sin\alpha\cos\beta, \qquad z = \sin\beta$$

waarbij α met de meridiaanlengte en β met de breedte correspondeert. Voor $\beta = 0$ krijgen we punten op de evenaar, voor $\beta = 90^\circ$ de noordpool en voor $\beta = -90^\circ$ de zuidpool.

Allereerst stellen we ons de opgave op het beeldscherm een afbeelding van de bol met een aantal breedtecirkels en meridiaancirkels te verkrijgen. De eenvoudigste methode is die waarbij punten (x,y,z) van de bol loodrecht op het beeldscherm geprojecteerd worden. Daarbij voeren we een tweede coördinatenstelsel in dat we vastgehecht denken aan het

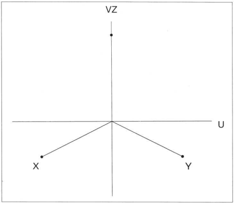
beeldscherm als projectievlak. De coördinaten noemen we u, v en w. Daarbij zijn u en v respectievelijk horizontale en verticale beeldschermcoördinaten en staat w loodrecht op het beeldscherm. Zonder bezwaar kunnen we de oorsprong van het u,v,w-stelsel met die van het x,y,z-stelsel laten samenvallen. De samenhang tussen de twee coördinatenstelsels is bepaald door een lineaire transformatie met een orthogonale coëfficiëntenmatrix. Ook die kiezen we zo eenvoudig mogelijk. We zorgen ervoor dat de z-as, de omwentelingsas van de aarde, geprojecteerd wordt als de v-as, en dat daarbij een helling van φ gekozen wordt. De keuze φ = 0 betekent dat de aardas evenwijdig is aan het projectievlak. Met $\varphi = 90$ kijken we bovenop de noordpool. Verder geven we de x-as en de y-as een symmetrische positie ten opzichte van het midden als geschetst in figuur 11.1.

De transformatieformules zijn dan

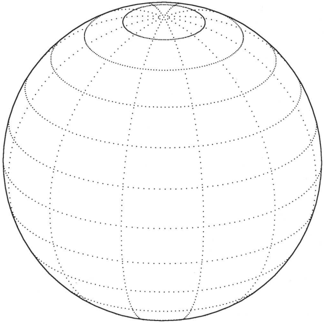
$$u = (-x + y)/\sqrt{2}$$
$$v = (-x - y)\sin\varphi/\sqrt{2} + z\cos\varphi$$
$$w = (x + y)\cos\varphi/\sqrt{2} + z\sin\varphi$$

Projecteren van een in x,y,z-coördinaten gegeven punt betekent niets anders dan dat we voor dat punt de coördinaten u,v moeten uitrekenen. De w-coördinaat hebben we eventueel nodig om te weten of het punt bij projectie zichtbaar is of niet. Een kleine meetkundige overweging maakt duidelijk dat $w > 0$ het zichtbare gebied van de bol is en $w < 0$ het onzichtbare.

Om als in het programma BOL een breedtecirkel te vormen geven we van de bolcoördinaten β een vaste waarde en laten we α lopen over een traject van 360° of 2π in radialen. In het programma kunnen we voor een staplengte van bijvoorbeeld 2° kiezen. Voor elke waarde van α berekenen we eerst x,y,z en vervolgens u,v,w. Wanneer $w > 0$ vormen we een punt op het beeldscherm, voor $w < 0$ onderdrukken we het plotten. Aldus kunnen we punt voor punt een willekeurige breedtecirkel vormen als een el-



figuur 11.1
Coördinatenstelsels, wereld en beeldscherm



figuur 11.2
Een bol met breedtecirkels en meridiaancirkels

lipsvormige stippellijn. Aan het begin van het programma hebben we de hellingshoek φ van de rotatieas de willekeurige waarde van 25° gegeven, maar het is leuk hetzelfde programma te herhalen met een andere waarde van φ. Overigens wordt in het programma met een lusstructuur een aantal breedtecirkels getekend. Op overeenkomstige wijze wordt een stelsel meridiaancirkels gevormd. Het resultaat is weergegeven in figuur 11.2.

De draaiende aarde

Vervolgens stellen we ons voor dat de bol met een constante snelheid om de z-as draait. In de formules kan dat worden bereikt door α te vervangen door $\alpha+t$ waarbij t een in hoekmaat uitgedrukte tijd is. Laten we t lopen van 0 tot 360 dan correspondeert dit met precies een omwenteling. In het programma AARDE kunnen we een willekeurig gebied om de aarde laten draaien. Gekozen is daartoe een denkbeeldig eiland dat de vorm heeft van een zeshoek, iets tussen Engeland en Australië in. Overigens kan met weinig moeite het programma aangepast worden voor uitbreiden voor een willekeurig gebied. Het is daarbij nodig de omtrek van het bewuste eiland of continent te benaderen door een gesloten veelhoek met, ter wille van de nauwkeurigheid, een flink aantal hoekpunten. Voor elk hoekpunt moeten dan de geografische lengte en breedte, uit de atlas

afgeleid, in een data-array worden meegegeven.

Tijdens de stapsgewijze draaiing dienen we telkens zowel over de oude posities als over de nieuwe posities van de hoekpunten van het eiland te beschikken. Daartoe reserveren we met de DIM-opdracht wat geheugenruimte. De beginposities, voor een echt eiland uit een atlas te halen, worden in de datalijst a(.),b(.) verenigd. Hoewel de computer intern in radialen rekent werken we zelf liever in graden. Daartoe moeten de door ons gebruikte hoeken in graden telkens omgerekend worden in radialen. Dat gaat met de formule:

$$180^\circ = \pi \text{ radialen}$$

Het getal π kent de computer niet en dat moeten we zelf in het programma declareren. Dat zou kunnen met een opdracht als PI = 3.141593, maar beter is het om PI = 4★ATN(1) te gebruiken, een wiskundige truc die berust op de eigenschap dat de tangens-functie van een hoek van $\pi/4$ radialen (45°) gelijk is aan 1. Met de teller n maken we in stapjes van 3° een dubbele omwenteling. De beweging van het eiland wordt geïmiteerd door naast elke positie telkens ook de volgende positie te berekenen. Nadat het eiland in de oude positie gewist is wordt het in de volgende positie opnieuw getekend. Bij een snelle computer geeft dat een illusie van een beweging. Bij film en televisie is het overigens niet veel anders. Wanneer de computer te langzaam is zou men kunnen proberen de stapgrootte bijvoorbeeld te verdubbelen tot 6°.

De beschreven projectiemethode van orthogonale projectie wordt veel gebruikt. Wanneer we tevreden zijn met een vlakke afbeelding van een klein gedeelte van een boloppervlak is die orthogonale projectie, bijvoorbeeld op het raakvlak in het centrum van het af te beelden gedeelte, heel effectief. De niet te vermijden vervormingen zullen pas storend zijn wanneer we ons verder van het raakpunt verwijderen.

Cilinderprojecties

Wil men meer, dan komen andere projectiemethoden in aanmerking. De wellicht meest voor de hand liggende methode om bijvoorbeeld de gehele aardbol af te beelden is de cilinderprojectie, waarbij men de bol omhuld denkt door een rakende cilinder. Nemen we aan dat de as van de cilinder samenvalt met de rotatieas van de aarde dan kunnen we punten (α,β) van de bol op een nader te bepalen methode op het cilinderoppervlak projecteren, waarna we de cilinder in een plat vlak afrollen.

In het eenvoudigste geval stemmen de Cartesische coördinaten (u,v) van de vlakke kaart precies overeen met de bolcoördinaten, dus

$$u = \alpha \quad \text{en} \quad v = \beta$$

De kaart ziet eruit als een liggende rechthoek met de verhouding $2\pi : 1$. Alles staat keurig op zijn plaats, maar de grootte klopt niet erg. Gebieden bij de equator zien er betrouwbaar uit, maar bij de polen is alles veel te groot. De kaart is dus niet vlaktrouw, dat wil zeggen de methode is niet

equivalent. Op een boloppervlak meten we afstanden doorgaans in hoekmaat, maar ook dat klopt niet, de cilinderprojectie is evenmin hoektrouw of conform. De beschreven methode staat bekend als de equidistante cilinderprojectie.

Een tweede voor de hand liggende methode is een horizontale meetkundige projectie waarbij de projecterende lijnen dus evenwijdig aan het equatorvlak lopen. Met een klein beetje schoolmeetkunde kunnen we al een opvallende ontdekking doen. Is R de straal van de aardbol, dan is het oppervlak van de bol gelijk aan $4\pi R^2$. Het oppervlak van de omhullende cilindermantel vinden we door de cilinder ergens langs een verticale lijn open te knippen en tot een platte rechthoek af te rollen. De afmetingen van de rechthoek zijn $2\pi R$ en $2R$. Het oppervlak is dus ook $4\pi R^2$, met andere woorden gelijk aan het aardoppervlak. Het ziet ernaar uit dat we een vlaktrouwe kaart gekregen hebben, en dat is ook zo, zoals een iets nauwkeuriger wiskundige analyse laat zien. De wiskundige formules zijn

$$u = \alpha \qquad \text{en} \qquad v = \cos(\beta)$$

De Mercator-projectie

We komen ten slotte tot de echte Mercator-projectie, een methode die noch vlaktrouw noch hoektrouw is, maar die wel een kaart levert die voor de scheepvaart heel handig is. De methode is bedacht door de zestiendeeeuwse Vlaamse kartograaf Gerard de Kremer (1512-1594); Mercator is de Latijnse vorm van zijn naam. Mercator was een veelzijdig geleerde met een grote wiskundige kennis en beschikkend over veelzijdige technische vaardigheden. In opdracht van Karel V vervaardigde hij een serie astronomische instrumenten. Voor een zeevaarder, en zeker voor een uit de tijd van Mercator, is het aantrekkelijk een vaste koers uit te zetten waarbij bijvoorbeeld meridiaanlijnen onder een vaste hoek worden gesneden. Het schip volgt dan een curve die bekend staat als een *loxodroom.* Voor kleine afstanden wijkt die curve niet zoveel af van de geodetische lijn, de kortste verbindingslijn tussen twee punten, op een boloppervlak een boog van een grote cirkel. Bij de Mercator-kaart correspondeert een loxodroom op de bol met een rechte lijn. De schipper heeft met andere woorden niet veel anders te doen dan op de kaart een rechte lijn uit te zetten. De Mercator-projectie geeft een geweldige vervorming in de buurt van de polen, maar zolang de zeevaart beperkt was tot gematigde breedten gaf dat niet zoveel problemen.

We weten niet hoe Mercator tot zijn kaartconstructie gekomen is. Vermoedelijk is hij uitgegaan van de hierboven besproken cilinderprojectie en heeft hij de kaart in de richting van de polen opgerekt totdat het klopte. Men spreekt aldus van de methode van de wassende breedten. Voor een hedendaags wiskundige met wat kennis van differentiaalrekening is het niet zo moeilijk de Mercator-projectie te herontdekken. Het is voldoende om uit te gaan van het gegeven dat de hoek die een willekeurige lijn met een breedtegraad maakt op de kaart even groot is. Mercator werkte dit kennelijk op een meetkundig constructieve wijze uit. Maar wij komen dan tot de onderstaande wiskundige formulering:

$$u = \alpha \quad \text{en} \quad v = \text{logtan}(\pi/4 + \beta/2)$$

Mercator zou deze formule niet hebben kunnen gebruiken, omdat de logaritmen in zijn tijd nog niet uitgevonden waren; dat zou pas in de zeventiende eeuw gebeuren. De Mercator-kaart is daarmee wel hoektrouw of conform, maar niet equivalent. Om een indruk van de grootte van bepaalde gebieden te krijgen is de Mercator-kaart nagenoeg onbruikbaar; een gebied als Groenland lijkt bedrieglijk groot als ware het een continent met de grootte van Afrika.

Verandering van grootte

Om een indruk te krijgen van de vertekening van de kaart kunnen we een leuke simulatie uitvoeren met behulp van het computerprogramma KAART. Maar eerst nog een kleine wiskundige overweging, die ons laat zien welke verandering van grootte van oppervlak kan optreden. Daartoe gaan we uit van een klein rechthoekig gebiedje op aarde, begrensd door de lengtegraden $\alpha, \alpha + d\alpha$ en de breedtegraden $\beta, \beta + d\beta$, een zogenaamd oppervlakte-element. Zouden we werken op een bol waarvan de straal de eenheid is en zouden we α en β in radialen rekenen, dan zou de grootte van dat rechthoekje praktisch gelijk zijn aan $\cos(\beta)d\alpha d\beta$, een formule die overeenkomstig de beginselen van de differentiaalrekening des te nauwkeuriger is naarmate de afmetingen van het rechthoekje kleiner zijn.

Voor de Mercator-kaart vinden we

$$du dv = d\alpha d\beta / \cos(\beta)$$

Dat scheelt met het overeenkomstige element op de bol een factor $\cos^2(\beta)$. Op een breedtegraad van 60° is dat al een factor 4 en dat wil zeggen dat een gebied er op de kaart al viermaal zo groot uitziet als het in werkelijkheid is.

Bij de allereerste door ons besproken kaartprojectie, $u = \alpha$ en $v = \beta$, is het verschil minder groot, een enkele factor $\cos(\beta)$. Op een breedte van 60° scheelt het nu een factor 2.

Computerexperimenten

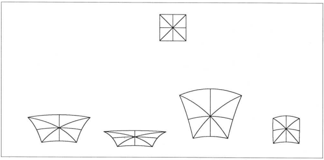
In het programma EILAND gaan we uit van een vierkant gebied, een eiland, gelegen in de buurt van de equator en bij de nul-meridiaan. We laten dat eiland in twaalf stappen in noordelijke richting opschuiven en bij elke stap laten we op het beeldscherm zien wat een van de besproken projectiemethoden ervan terechtbrengt. Dat opschuiven van het vierkant moeten we ons voorstellen als een rotatie van het aardoppervlak om een horizontale as in het equatorvlak. Het vierkant blijft dus onder alle omstandigheden een vierkant, een figuur die congruent is met die in de oorspronkelijke positie. We zien dan, afhankelijk van de gekozen methode, vreemde en wellicht onverwachte vervormingen, vooral wanneer het vierkant zich op hogere breedtegraden bevindt. Voor een deel komt dat

doordat een breedtecirkel geen grote cirkel is en dus geen kortste verbinding tussen twee punten levert. De horizontale zijden van het vierkant krijgen derhalve een steeds grotere kromming naarmate we op hogere breedtegraden komen. De verticale zijden van het vierkant gaan zich bij de opschuiving over een telkens groter gebied van lengtegraden uitstrekken en dat levert bij projectie soms een soort trapeziumachtige vertekening met kromme zijden. Men moet het zelf op het computerscherm zien om te begrijpen hoe bedrieglijk een kaart kan zijn.

Het programma is zodanig geschreven dat men het met weinig moeite kan uitbouwen of toepassen op andere situaties. De projectieformules vindt men aan het einde als eenregelige subroutines. Met het keuzegetal p kan men zelf de projectiemethode uitkiezen. Naast de genoemde drie projectiemethoden is nog de zogenaamde methode van Sanson-Mercator opgenomen, die niet conform maar wel equivalent is. Het eiland is vastgelegd met de α,β-coördinaten van de hoekpunten. In het programma vullen we die punten aan door interpolatie met op elke zijde nog een extra aantal, het getal n gekozen als 32. Aldus kan het vierkant in elke ligging uit $4n$ losse puntjes worden getekend. Het programma kan natuurlijk nog op allerlei wijzen uitgebreid en gevarieerd worden, maar dat laten we aan de inventiviteit van de lezer over.

Gebruik makend van een dergelijke variant geven we in figuur 11.3 een overzicht van wat we met een paar kaartprojecties te zien krijgen bij een vierkant land waarvan het centrum op de zestigste breedtegraad geplaatst is. Het land denken we gelegen tussen 50° en 70°, maar dat betekent niet dat zuid- en noordgrens ook de bijbehorende breedtecirkels zijn. De werkelijke grenzen zijn bogen van grote cirkels en die worden bij de kaartprojecties meestal als gebogen lijnen afgebeeld. Ook bij de oost- en de westgrens van het vierkant kunnen we op een kaart vreemde vervormingen verwachten. Immers, naarmate we de polen naderen correspondeert een zelfde afstand in lengtegraden met een steeds kleinere afstand in geodetische zin gemeten langs een grote cirkel.

We nodigen de lezer uit aan de hand van het programma KAART zelf experimenten uit te voeren. Met wat geduld kan hij dan een land – dat in grootte en vorm met bijvoorbeeld Australië overeenstemt – op een willekeurige plaats op aarde neerzetten en er met een willekeurige projectie een kaart van maken.



figuur 11.3
Een vierkant land en wat de kartograaf ervan maakt

12 Tussen de sterren

Een van de grootste ontdekkingen in de wetenschap is wel dat de beweging van alle hemellichamen, sterren, planeten en satellieten bepaald is door slechts een enkele wet. Die ontdekking werd gedaan door Newton en werd het begin van een nieuwe tak van wetenschap, de hemelmechanica. Newtons wet luidt in woorden dat het produkt van massa en versnelling van een enkel hemellichaam gelijk is aan de som van de krachten die erop werken.

De enige kracht in de mechanische kosmos is de zwaartekracht. Elke massa, waar ook in het heelal aanwezig, oefent op elke andere massa een aantrekkende kracht uit. Voor twee puntvormige hemellichamen P_1 en P_2 is de kracht gericht langs de verbindingslijn evenredig met de massa's van de twee lichamen en omgekeerd evenredig met het kwadraat van de onderlinge afstand. De evenredigheidsconstante is een natuurconstante waarvan de grootte afhangt van de eenheden waarmee we werken.

Strikt genomen geldt de wet van Newton voor zogenaamde massapunten of puntmassa's, denkbeeldige objecten waarvan de afmetingen kunnen worden verwaarloosd, maar die wel een bepaalde massa bezitten. Bewezen kan worden dat een bolvormig lichaam waarvan de massa homogeen verdeeld is zich gedraagt als een massapunt in het middelpunt waarin alle massa is opgehoopt. Om die reden kan de wet van Newton met groot gemak worden toegepast op de onderlinge beweging van bijvoorbeeld twee sterren. In het wiskundige model dat we aanstonds bespreken vatten we de sterren dan op als twee bewegende punten P_1 en P_2 met respectievelijk de massa's m_1 en m_2. Andere aantrekkende objecten zijn er niet in dat model. De theorie van Newton zegt dat we net kunnen doen alsof het zwaartepunt O van de twee sterren op zijn plaats blijft en dat de beweging van de sterren om dat gemeenschappelijke centrum in een plat vlak plaatsvindt, een vlak dat we als het coördinatenvlak x,y kunnen opvatten.

De wet van Newton kunnen we dus toepassen op elk van de twee sterren. In de wiskundige formulering van deze wet moeten we bedenken dat begrippen als snelheid, versnelling en kracht de betekenis hebben van een vector: een grootheid gekenmerkt door grootte en richting. Omdat het prettig is daaraan ook een aanschouwelijke interpretatie toe te kennen wordt een vector vaak geïnterpreteerd als een pijltje. Om ermee te kunnen rekenen, en dat is nodig voor het opstellen van een computerprogramma, werken we met de componenten van een vector, diens projecties op de assen van een coördinatenstelsel. De wiskundige formulering is

(12.1) $$m_1 \mathrm{d}^2\mathbf{r}_1/\mathrm{d}t^2 = -Gm_1m_2(\mathbf{r}_1 - \mathbf{r}^2)/D^3$$

waarin de gebruikte symbolen de volgende betekenis hebben: $\mathbf{r}_1$ en $\mathbf{r}_2$ zijn de vectoren OP_1 en OP_2 die de posities van de twee sterren bepalen, D is gelijk aan de onderlinge afstand P_1P_2 van de twee sterren, G is de constante van de zwaartekracht (gravitatie).

Voor de tweede ster geldt een overeenkomstige vergelijking, maar nieuwe informatie geeft dat niet omdat het zwaartepunt van de twee puntsterren steeds met de oorsprong samenvalt, in formulevorm

$$(12.2) \qquad m_1\mathbf{r}_1 + m_2\mathbf{r}_2 = 0$$

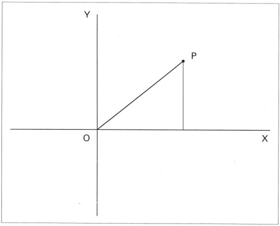
Van Newton tot Kepler

De differentiaalvergelijking (12.1) kan langs wiskundige weg geïntegreerd worden. Dat wil in dit geval zeggen dat we in staat zijn voor $\mathbf{r}_1$ en $\mathbf{r}_2$ eenvoudige formules af te leiden met behulp waarvan de posities van beide sterren op elk willekeurig tijdstip berekend kunnen worden. Omdat de wiskundige details meer in een wiskundeboek thuishoren beperken we ons hier tot de grote lijnen, teneinde de lezer een indruk te geven van de wijze waarop uit de wet van Newton de wetten van Kepler, zijn beroemde voorganger, kunnen worden afgeleid.

Die afleiding gaat het beste in zogenaamde poolcoördinaten r en φ die in figuur 12.1 nog even zijn samengevat.

Ten opzichte van een Cartesisch coördinatenstelsel OXY kunnen de poolcoördinaten als volgt beschreven worden. Is P een punt met de poolcoördinaten r en φ, dan is r gelijk aan de afstand OP en φ gelijk aan de hoek die de gerichte lijn OP maakt met de positieve x-as. Zijn x en y de Cartesische coördinaten van P dan gelden de omrekeningsformules

$$(12.3) \qquad x = r\cos\varphi, \qquad y = r\sin\varphi$$



figuur 12.1
Poolcoördinaten en Cartesische coördinaten

De integratie van de bewegingsvergelijkingen leidt dan tot de volgende resultaten

(12.4) $r_1 = a_1/(1+e\cos(\varphi)$

en

(12.5) $r_1^2 d\varphi/dt = b_1$

Die formules kunnen we interpreteren als wetten van Kepler.

De formule (12.4) bepaalt de vorm van de baan van de eerste ster, en dat is een kegelsnede waarvan de oorsprong, het gemeenschappelijke zwaartepunt, een brandpunt is. De baan van de tweede ster is eveneens een kegelsnede die gelijkvormig is met de baan van de eerste ster.

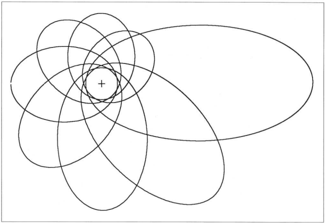
De parameter e, een niet-negatief getal, staat bekend als de *excentriciteit*. Is $e = 0$ dan is $r_1 = 0$ zodat de baan van de eerste ster een cirkel is. Die van de tweede ster is een cirkel met hetzelfde middelpunt, en de verbindingslijn van beide sterren gaat steeds door dat middelpunt. Is $0 < e < 1$ dan kunnen we zonder moeite uit (12.4) afleiden dat de baan in elk geval begrensd is. Een cosinusfunctie neemt immers slechts waarden tussen -1 en 1 aan zodat de noemer in het rechterlid van (12.4) nooit nul kan worden. De baan van de eerste ster is dan een ellips met halve lange as $a_1/(1\text{-}e)$ en halve korte as $a_1/(1\text{-}e)$. De baan van de tweede ster is weer een daarmee gelijkvormige ellips. Is $e = 1$ dan wordt de noemer van (12.4) nul voor $\varphi = \pi$ zodat r_1 oneindig wordt. De sterren hebben dan een parabool als baan waarvan de oorsprong weer het gemeenschappelijke brandpunt is. Is $e > 1$ dan is (12.4) slechts bruikbaar voor die waarden van φ waarvoor de noemer positief is. De baan is dan een boog van een hyperbool. De sterren naderen elkaar vanuit de verte op de wijze van een bijna-botsing. Dicht bij elkaar gekomen ($\varphi = 0$) draaien ze weer van elkaar weg om nooit meer terug te komen.

De tweede vergelijking (12.5) is de wiskundige vorm van de bekende *perkenwet* van Kepler. Terwijl de eerste formule ons alleen de geometrie van de baan geeft, geeft de tweede formule ons de mogelijkheid aan de positie van een ster een tijdstip te verbinden.

Een planeet om de zon

De gevonden resultaten kunnen we toepassen op de beweging van een planeet om de zon. Omdat de massa van een planeet praktisch te verwaarlozen is ten opzichte van die van de zon kunnen er een paar vereenvoudigingen worden aangebracht. Het zwaartepunt van het stelsel zon-planeet valt dan nagenoeg samen met het middelpunt van de zon. In het mathematische model kunnen we de zon dan als een vast punt, de oorsprong O van een x,y-coördinatenstelsel, beschouwen. De baan van de planeet wordt dan gegeven door

$$r = a/(1 + e\cos(\varphi))$$



figuur 12.2
Een achttal planetenbanen

Voor $\varphi = 0$ is de voerstraal zo klein mogelijk. De planeet staat dan zo dicht mogelijk bij de zon en de afstand bedraagt $a/(1+e)$. Voor $\varphi = \pi$ heeft de planeet de grootste afstand tot de zon en die is dus $a/(1\text{-}e)$. De som van beide grootheden, namelijk $2a/(1\text{-}e^2)$, is de grootste diameter van de planetenbaan. De tweede vergelijking

$$r^2 d\varphi/dt = \text{constant}$$

geeft de mogelijkheid om aan de baan een tijdschaal te verbinden. In het programma PLANEET geven we een aantal planetenbanen die op deze twee vergelijkingen gebaseerd zijn. Er worden daarin een achttal planetenbanen berekend en afgebeeld. De excentriciteit neemt telkens een beetje af. We beginnen met een sterk ellipsvormige baan en eindigen met een baan die bijna een cirkel is (figuur 12.2).

De reis van een ruimtesonde

We lezen vaak dat er een ruimtesonde gelanceerd is met als doel een of meer verre planeten te verkennen. Een onderneming waaraan behalve technisch vernuft ook heel nauwkeurig rekenwerk ten grondslag dient te liggen. Het kost nogal wat energie om het ruimteschip de aantrekkingskracht van de aarde te laten overwinnen, en er is maar weinig energie over om onderweg wat bij te sturen. We lezen dan dat de baan zo gekozen is dat de satelliet na het verlaten van de aarde door een planeet op dusdanige wijze wordt aangetrokken dat die er als het ware wat extra energie aan toevoegt, zoals een kogelslingeraar met een extra draai van zijn lichaam een kogel ver weg weet te slingeren.

Inderdaad staan de wetten van de hemelmechanica ons toe een klein beetje energie van een planeet af te tappen. De kunst is alleen dat goed te

doseren, want een teveel doet het ruimtetuig voor eeuwig in de diepten van het heelal verdwijnen. Het is dus zaak de sonde op de juiste wijze langs een planeet te laten scheren. Aan dit probleem wijden we een tweetal simulatieprogramma's, SONDE en STARSHIP. Het programma SONDE betreft een eenvoudige wiskundige simulatie die niet helemaal met de wetten van de hemelmechanica in overeenstemming is, maar die wel duidelijk laat zien hoe een ruimtesonde op zijn reis door de aantrekkingskracht van twee puntvormige sterren wordt beïnvloed. Het programma STARSHIP is iets ingewikkelder, maar dan ook geheel in overeenstemming met de astronomische realiteit. Bij de beschrijving van dat programma gaan we ook iets meer in op de achterliggende wiskunde.

In het programma SONDE veronderstellen we dat een sonde, waarvan de massa te verwaarlozen is, een beweging in een plat vlak uitvoert en daarbij aangetrokken wordt door twee stilstaande puntvormige sterren in de posities $(a,0)$ en $(-a,0)$. De bewegingswetten van Newton worden daarbij correct geïnterpreteerd. De enige 'fout' die we maken is dat we aannemen dat de twee sterren ten opzichte van elkaar stilstaan. In werkelijkheid kan dat natuurlijk niet. Of ze zouden zich naar elkaar toe bewegen, of ze zouden volgens ellips- of cirkelvormige banen om elkaar heen draaien. Toch is het dynamische systeem niet in strijd met de principes van de mechanica. Het is alsof de sterren verbonden zijn door een massaloze staaf die de beweging van de sterren verhindert. Het model laat dan ook alles zien wat we zouden kunnen verwachten. De sonde kan soms vele malen om de twee sterren heen draaien, buitenwaarts en binnenwaarts. Soms kan het gebeuren dat de sonde, te dicht bij een van de twee sterren gekomen, er zoveel energie bij krijgt dat hij uit het systeem wordt gegooid. Het programma laat een groot aantal mogelijkheden toe. De beginpositie van de sonde kan aan het begin van het programma op willekeurige wijze ingevoerd worden, als een punt op een cirkel met gegeven straal en gegeven fase. Aangenomen is dat de beginsnelheid gericht is volgens een raaklijn aan de cirkel – dat zou ook anders kunnen – maar dat de grootte daarvan ook via het toetsenbord ingebracht kan worden. Verder bevat het programma nog de constante c als een soort gravitatieconstante, en de tijdstap h. Grotere waarden van h laten het programma sneller verlopen, maar dat kan wel ten koste gaan van de nauwkeurigheid.

Het programma STARSHIP

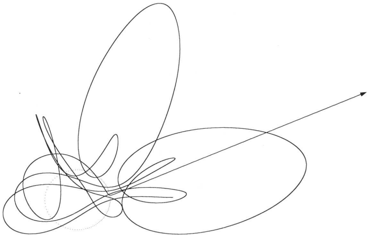
In dit programma gaat het eigenlijk om een speciaal geval van het zogenaamde drielichamenprobleem, een boeiend onderdeel van de hemelmechanica waarover men bijvoorbeeld in het boek *Dynamische systemen en chaos* van Broer en Verhulst nadere informatie kan vinden. We stellen ons voor dat ons ruimtevaartuig een bezoek brengt aan een dubbelster waarvan de componenten gelijke massa's hebben en in een elliptische baan om hun gemeenschappelijk zwaartepunt bewegen. Het ruimteschip laten we bewegen in het vlak van het sterrenstelsel. Dat doen we natuurlijk ter vereenvoudiging, om de berekeningen niet al te ingewikkeld te maken. Beginpositie en beginsnelheid kunnen we in het programma willekeurig kiezen, uiteraard weer in het vlak van beweging.

De beweging van de sonde is bepaald door de tijdafhankelijke vector $\mathbf{r}_3(t)$ waarvan we hier maar twee componenten nodig hebben, in het programma aangeduid met x en y. Volgens Newton voldoet de versnelling $d^2\mathbf{r}_3/dt^2$ van de sonde aan de vectoriële bewegingsvergelijking (zie bijvoorbeeld Broer en Verhulst, blz. 199)

$$m_3 d^2\mathbf{r}_3/dt^2 = -Gm_1m_3(\mathbf{r}_3 - \mathbf{r}_1)/P_{13}^3 - Gm_2m_3(\mathbf{r}_3 - \mathbf{r}_2)/P_{23}^3$$

waarbij $\mathbf{r}_1$ de positievector is van het eerste hemellichaam, in coördinaten (x_1, y_1) enzovoort. Met P_{13} is de lengte van de vector P_1P_3 bedoeld. We nemen als gezegd m_1 en m_2 gelijk aan 1, geen essentiële vereenvoudiging overigens. De waarde van de massa m_3 hebben we niet nodig. De bovenstaande vergelijking drukt uit dat de versnelling die de sonde op een bepaald moment ondervindt afhangt van de aantrekking van elk van de twee afzonderlijke sterren. In het computerprogramma kunnen we natuurlijk geen vectoren gebruiken en moet alles gesplitst worden in x- en y-componenten.

De overeenkomstige bewegingsvergelijkingen van de twee sterren hoeven we niet te gebruiken, omdat ze volgens de wetten van Kepler in een elliptische baan om elkaar heen cirkelen. Het wiskundige materiaal voor het computerprogramma is nu compleet en er rest ons 'slechts' de differentiaalvergelijkingen van de ruimtesonde te integreren. We doen dat op de meest eenvoudige wijze door met heel kleine stapjes h in de tijd verder te gaan. Het gaat ons er in eerste instantie toch om het computerprogramma eenvoudig en snel te houden zodat er op het scherm, ondanks de overigens zeer geringe onnauwkeurigheid, een goed beeld ont-



figuur 12.3 Het bezoek van een ruimtesonde aan een dubbelster

staat van wat we in de ruimtevaart kunnen verwachten. Verdere details diene men in het programma zelf af te lezen. Er kan heel wat mee geëxperimenteerd worden. Het is interessant om telkens heel kleine veranderingen in de beginsituatie aan te brengen, en dan te zien wat er op den duur gebeurt. Het zal de experimentator wellicht opvallen dat er weer weinig voor nodig is om het ruimtescheepje door de aantrekkende dubbelster de ruimte ingeslingerd te zien.

Over het programma kunnen nog een paar opmerkingen worden gemaakt. De kern is de voorwaardelijke lus. Het programma stopt wanneer een willekeurige toets wordt aangeraakt of wanneer de sonde het beeldscherm afvliegt. De twee sterren worden afgebeeld als bewegende punten door eerst de getoonde stippen te verwijderen en de nieuwe posities als (bijvoorbeeld) gele stippen weer te geven. Voor de aangegeven waarden van de parameters ziet de baan van de sonde eruit als in figuur 12.3. Het blijkt dat de satelliet een vrij groot aantal malen om en tussen de sterren door beweegt. Ten slotte komt de sonde zo dicht bij een van de twee sterren, dat hij een opzwieper krijgt die hem het systeem definitief doet verlaten. Wie het op de computer ziet gebeuren zal wel merken dat heel kleine wijzigingen in de beginparameters op den duur dramatische gevolgen kunnen hebben. De banen blijken dan ook vaak alle kenmerken van chaos te bezitten. Wie een niet zo snelle computer heeft kan door het kiezen van een grotere waarde van de tijdstap *h* het proces aanmerkelijk versnellen.

13 Maak uw eigen planetarium

Wanneer ons wordt gevraagd namen te noemen van grote Nederlanders uit voorbije eeuwen, dan denken we wellicht aan de vertegenwoordigers van de schilderkunst, met Rembrandt als de meest beroemde. Vermoedelijk zal iemand als Christiaan Huygens (1629-1695) niet of weinig worden genoemd, hoewel hij voor de wetenschap nauwelijks minder belangrijk was dan Isaac Newton, de grondlegger van de moderne mechanica. Huygens hield zich onder andere bezig met de optica, de theorie van de voortplanting van lichtstralen. Zoals we al in hoofdstuk 12 hebben gezien heeft Newton ons theoretisch inzicht verschaft in de beweging van de hemellichamen. Maar Huygens keek er ook naar met door hem zelf ontworpen kijkers en samen met zijn broer sleep hij de benodigde lenzen. En zo staat Huygens ook bekend als de ontdekker van de ring van Saturnus (1655).

Een eeuw later waren de ontdekkingen van Huygens en anderen gemeengoed geworden. Men kon goede kijkers en microscopen maken, en ook nauwkeurig lopende klokken. Hoe de belangstelling voor de wetenschap ook onder het 'gewone publiek' leefde blijkt onder andere uit de grote populariteit van zogenaamde planetaria, mechanische simulaties van ons zonnestelsel. Omstreeks 1775 bouwde een Friese wolkammer, Eise Eisinga, een dergelijk model dat we nog steeds in Franeker kunnen bewonderen. Hangend aan het plafond van een kamer bewegen kleine balletjes zich als miniatuurplaneten om de zon. Ingewikkelde raderwerken als van een gecompliceerde klok zorgen voor een nauwkeurige beweging, zodat het model steeds een getrouwe afspiegeling van de werkelijkheid is.

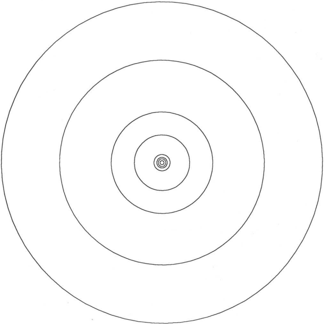
Het planetenstelsel gesimuleerd

Voor het maken van een planetarium hoeft men niet meer te beschikken over de kennis en de vaardigheid van een klokkenmaker. Elektronica kan de raderwerken vervangen, en een trillend kristal maakt de slinger van Huygens overbodig. Het is zelfs mogelijk een planetarium te maken zonder enige technische kennis van elektronica te bezitten. Met een computer en een beetje software kan alles op verregaande wijze gesimuleerd worden. In dit hoofdstuk laten we aan de hand van een paar eenvoudige programma's zien hoe dat kan. We beschouwen daartoe een paar concrete deelproblemen, die men zou kunnen zien als bouwstenen van een realistisch computer-planetarium. We beginnen met het maken van een simulatie van het zonnestelsel.

Met behulp van het programma PLANEET1 laten we op het beeld-

scherm acht planeten inclusief de aarde om de zon over cirkelvormige banen heen draaien. Slechts Pluto laten we weg, maar die wordt toch door de astronomen niet als een echte planeet beschouwd. De stralen van de cirkels zijn op de juiste schaal, en ook de snelheden waarmee de (puntvormige) planeten bewegen zijn evenredig met de werkelijke snelheden in het zonnestelsel. De aanvangsposities van de planeten zijn in dit programma op toevallige wijze gekozen, maar met weinig moeite zou men het programma aan bijvoorbeeld de huidige astronomische realiteit kunnen aanpassen. Het programma spreekt verder voor zich. Aan het begin kan men door middel van de schaalfactor kiezen voor het totale beeld ($s = 1$) of voor bijvoorbeeld het gedeelte met Jupiter als buitenste planeet ($s = 6$). Met een waarde voor a kan men de tijdsduur in aantal jaren instellen. Na a omlopen van de aarde stopt het programma dan automatisch. De planeten zijn weergegeven als kleine gekleurde cirkelvormige schijfjes. De beweging wordt geïmiteerd door van elke planeet eerst de huidige positie te verwijderen door hetzelfde schijfje de kleur van de achtergrond te geven en vervolgens het schijfje in de oorspronkelijke kleur in de nieuwe positie aan te brengen (zie figuur 13.1).

Wie dit programma op een moderne computer wil uitvoeren kan het als een probleem ervaren dat de beweging te snel of te langzaam plaats-



figuur 13.1 Planetenbanen op schaal in ons zonnestelsel

vindt. De eenvoudigste oplossing is om de constante h in het programma een kleinere of grotere waarde te geven. Een andere mogelijkheid om een willekeurig programma iets te vertragen is het aanbrengen van een 'pause'-opdracht waardoor het programma telkens een voorgeschreven aantal (milli)seconden stilstaat. Helaas bestaat daarvoor in de gebruikelijke programmeertalen geen eensluidende benaming. Het kan bijvoorbeeld 'SLEEP' zijn (bij QBasic alleen in gehele seconden) of 'DELAY'. Deze discrepantie is de reden dat dergelijke opdrachten in de hier gegeven programma's niet gebruikt worden. We kunnen ons ook behelpen met een zogenaamde 'dummy lus', een 'FOR...NEXT'-opdracht die niets doet maar alleen wat tijd kost.

Wie de planetenschijfjes over het beeldscherm ziet bewegen zal allicht verrast zijn dat de buitenplaneten in relatieve maat zo langzaam bewegen ten opzichte van de binnenplaneten. De verklaring voor dat verschijnsel ligt besloten in de derde wet van Kepler: 'De kwadraten van de omlooptijden zijn evenredig met de derde machten van de lange assen van de ellipsbanen'. Wanneer de ellipsvormige baan een cirkel is moet de 'lange as' natuurlijk worden opgevat als de straal van de cirkel. In het programma is van deze wet dan ook gebruik gemaakt. Het betekent bijvoorbeeld dat een planeet als Neptunus met een baan van dertig AE (de afstand van de aarde tot de zon als astronomische eenheid) een omlooptijd heeft van $30^{1.5}$ of 164.8 jaar.

In werkelijkheid zijn de banen van de planeten geen echte cirkels maar ellipsen. Overigens wijken die ellipsen zo weinig af van cirkels dat men op het beeldscherm nauwelijks verschil merkt. Ook liggen de banen van de planeten met de zon niet in hetzelfde vlak, maar ook dat scheelt niet zoveel. Zouden we het zonnestelsel waarnemen vanuit een buitenaards ruimteschip dat zich op enige afstand van de zon loodrecht op het (bijna) gemeenschappelijke vlak, het zogenaamde middenvlak, bevindt, dan zouden we precies datgene zien wat we zojuist op het computerscherm hebben gesimuleerd.

Dwalen langs de hemel

Het woord planeet staat in verband met een Grieks werkwoord dat dwalen of zwerven betekent. Een planeet noemde men dus een dwaalster, en inderdaad blijken de planeten aan de hemel merkwaardige banen te volgen, heen en weer met platte lussen en af en toe een schijnbare stilstand. Voor de Griekse geleerden aan het begin van onze jaartelling een mysterie met een moeizame theoretische verklaring, de zogenaamde epicykeltheorie van Ptolemaeus, maar voor ons met een beetje kennis van stereometrie en goniometrie gemakkelijk te begrijpen.

We stellen ons tot taak die schijnbare beweging op het beeldscherm te simuleren. Het is voldoende wanneer we ons hierbij tot een enkele planeet, Jupiter bijvoorbeeld, beperken. Essentieel is nu dat het vlak van beweging van Jupiter niet met dat van de aarde samenvalt, maar er juist een kleine hoek, de zogenaamde inclinatie, mee maakt. Wel benaderen we in het nieuwe model de banen van aarde en Jupiter door cirkels.

Als aardse waarnemers die naar de hemel kijken beginnen we met de

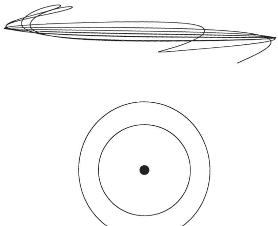
baan van de aarde zo eenvoudig mogelijk te kiezen, en wel als een cirkel met straal 1 in een Cartesisch coördinatenstelsel:

$$x = \cos(t), \qquad y = \sin(t), \qquad z = 0$$

De baan van Jupiter denken we bepaald door een grotere cirkel waarvan het vlak een kleine hoek φ maakt met het vlak van de aardbaan:

$$x_1 = r\cos(\varphi)\cos(ct), \qquad y_1 = r\sin(ct), \qquad z_1 = -r\sin(\varphi)\cos(ct)$$

De constante c is de hoeksnelheid waarvan de waarde bepaald is door de derde wet van Kepler als $c = r^{-1.5}$. Wat we vanuit de aarde aan de hemel waarnemen is het oneindig verre punt van het lijnstuk PP_1, diens richting dus, waarbij P de aarde en P_1 de planeet is. Het komt er dus op neer op de een of andere manier die richting als punt op het beeldscherm weer te geven. In het programma PLANEET2 is het een en ander gerealiseerd. De banen van aarde en Jupiter zijn zichtbaar gemaakt als cirkels in het x,y-vlak waarbij nu de inclinatie van de baan van Jupiter is verwaarloosd. De vector PP_1 wordt berekend en vervangen door een eenheidsvector, omdat van die vector alleen de richting van belang is. Van die vector wordt aan de bovenkant van het beeldscherm de projectie op het x,z-vlak als punt afgebeeld. Het is dus alsof we in het vlak van de aardbaan naar die vector kijken. Door de beweging van aarde en Jupiter beweegt die projectie heen en weer. Zou de baan van Jupiter met die van de aarde samenvallen, dan zou die heen en weer gaande beweging volgens een rechte lijn gebeuren, niet zo interessant. Maar juist omdat de baan van Jupiter een kleine incli-



figuur 13.2 De schijnbare baan van een planeet

natie heeft, beweegt het beeldpunt zich ook een beetje naar boven en naar beneden, en vormen zich lussen. Veel mooier, en dat is precies wat een hemelbeschouwer waarneemt bij het observeren van de planeten. Overigens is bij de projectie nog een klein beetje schaling toegepast waarvan het enige doel het verkrijgen van een wat beter plaatje is. De totale tijdsduur is heel willekeurig op vijftig radialen, dus zowat acht jaar, gesteld (zie figuur 13.2).

Kometen jagen

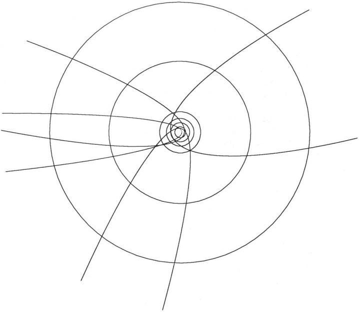
Van oudsher plachten kometen de mensheid schrik en ontzag in te boezemen. Oude verhalen, schilderingen en gravures getuigen daarvan. Misschien maakte eens de inslag van een komeet een einde aan het tijdperk van de dinosauriërs. Feit is dat aarde en maan in hun bestaan vele inslagen gekend hebben van kleine en grote brokken ruimtelijke materie. Men denkt dat ver buiten de omloop van de laatste planeet een ringvormige wolk oermaterie aanwezig is, de zogenaamde Oort-wolk, zo genoemd naar de Leidse astronoom Jan Oort. Die wolk is als het ware de kraamkamer van de kometen. Normaal bewegen die deeltjes zich langs ongeveer dezelfde cirkelvormige banen om de zon. Bij deeltjes moet men denken aan alles tussen minuscule zandkorreltjes en grote rotsblokken in, aan bevroren gas en water, aan ijzer en koolstof, aan een vuile sneeuwbal pleegt men te zeggen. Maar alle, groot of klein, gehoorzamen ze aan de wetten van de mechanica, aan die van Newton en van Kepler. Soms treedt er een storing op, misschien door het langstrekken van een passerende naburige ster, en dan kan het gebeuren dat deeltjes zich losmaken uit de stofring en zich in de richting van de zon gaan bewegen. Dat zou de geboorte van een nieuwe komeet kunnen zijn. Zo'n komeet volgt ook weer een Kepler-baan, meestal een ellips met een grote excentriciteit. De komeet kan op zijn weg heel dicht bij de zon komen en kan daarbij ook dicht in de buurt van de aarde of de binnenplaneten komen. Er bestaat altijd een kans dat de aarde een voltreffer krijgt. Waarschijnlijk is het weliswaar niet, maar evenmin is het uitgesloten. Het zou voor kleine komeetjes best wel eens eenmaal in de honderdduizend jaar kunnen gebeuren, en dat zo'n natuurgebeuren grote gevolgen kan hebben laat zich denken. Zolang de komeet nog ver weg is zien we er niets van, omdat de komeet zelf te klein is om voldoende zonlicht te reflecteren. Maar zodra de komeet in de buurt van de zon komt kan zich de karakteristieke heldere staart ontwikkelen, waaraan de naam 'komeet' ontleend is.

Er zijn heel wat amateurs die 's nachts de hemel afzoeken in de hoop een nieuwe komeet te ontdekken. Niet zozeer om de eerste te zijn met het brengen van een onheilsboodschap, maar eerder om de sport en de sensatie. Bovendien draagt een nieuwe komeet voortaan de naam van de eerste ontdekker(s). Met uw computer bent u zelf zo'n ontdekker, en achter uw computer zittend kunt u de ene na de andere komeet op uw beeldscherm oproepen en bestuderen. Weliswaar verschaft dat geen internationale naamsbekendheid, maar iets van de sport en de sensatie is gebleven, zeker voor de computeraar die over enige fantasie beschikt. Met het programma KOMEET wordt op het scherm het planetenstelsel, geheel of

gedeeltelijk, zichtbaar als een stelsel cirkels. Ergens daarbuiten bevindt zich als op een onzichtbare cirkel de kraamkamer van de kometen. Telkens laten we daar in een toevallig gekozen positie een komeet ontstaan die vervolgens een ellipsvormige baan om de zon gaat uitvoeren. De excentriciteit van de baan kiezen we eveneens op toevallige wijze. De ellips kunnen we in poolcoördinaten voorstellen door de vergelijking

$$r = R(1 - e)/(1 - e\cos(\varphi))$$

waarbij R de straal van de Oort-wolk en e de excentriciteit is. De fase laten we overeenkomstig de perkenwet met kleine stapjes toenemen totdat een omloop voltooid is. Het programma is zodanig gemaakt dat er telkens een nieuwe komeet gecreëerd wordt totdat met een willekeurige toetsaanslag het programma wordt gestopt (figuur 13.3).



figuur 13.3 Bezoek van vier kometen

14 Relaxatietrillingen

Trillingsverschijnselen in de natuurwetenschappen kunnen op verschillende wijzen gemodelleerd worden. Het eenvoudigste model is de harmonische trilling, die eendimensionaal door $u = a \cos \sigma t$ kan worden beschreven. We kunnen u opvatten als de uitwijking van een deeltje uit de evenwichtsstand en t als de tijd in een geschikte maat, seconden bijvoorbeeld. De constante σ is de frequentie, waarbij $2\pi/\sigma$ de trillingstijd is, de tijd die nodig is om precies een enkele trillingsgolf uit te voeren. Het aantal trillingen per seconde is dus $\sigma/2\pi$.

Gedempt en ongedempt

Soms is dat eenvoudige model slechts bij benadering juist, bijvoorbeeld omdat er in de praktijk een (kleine) dempende kracht aanwezig is. In een dergelijke situatie verdwijnt uit het systeem energie. Door wrijving kan mechanische energie verloren gaan, een effect dat men dissipatie van energie noemt.

Een harmonische trilling, al of niet gedempt, hoort bij lineaire modellen en is bepaald door een lineaire differentiaalvergelijking. Een dergelijke vergelijking kan vaak met eenvoudige analytische hulpmiddelen, in formulevorm, worden opgelost. Het prototype van de ongedempte variant is de differentiaalvergelijking

(14.1) $\quad d^2u/dt^2 + \sigma^2 u = 0, \qquad \sigma > 0$

waarvan de algemene oplossing kan worden geschreven als

$$u = a \cos(\sigma t + \alpha)$$

waarbij de maximale uitwijking a, de amplitude, en de fase α willekeurige waarden kunnen aannemen. In plaats van de cosinus hadden we ook de sinusfunctie kunnen nemen, een zaak van conventie of gewoonte.

Nog even een opmerking over de gebezigde notatie. Uiteraard hangt u van t af; met u bedoelen we dus de waarde van die functie op het tijdstip t. Een uitvoeriger notatie zou $u(t)$ zijn, maar om praktische redenen maken we vaak van de verkorte notatie gebruik wanneer er geen misverstand is te duchten.

Een gedempte harmonische trilling wordt analoog beschreven door de differentiaalvergelijking

(14.2) $\quad d^2u/dt^2 + \mu du/dt + \sigma^2 u = 0$

waarbij de term met de eerste afgeleide de mate van demping aangeeft. De (positieve) coëfficiënt μ geeft de mate van demping aan. Ook daarbij zijn de trillingen in principe sinusvormig, maar de amplitude neemt exponentieel af volgens de factor $\exp(-\mu t/2)$.

Numerieke integratie

Hoewel het mogelijk – en zelfs vrij eenvoudig – is om de vergelijking (14.2) met behulp van formules op te lossen, maken we ter wille van een computersimulatie vooreerst gebruik van een numerieke benadering, een zogenaamde numerieke integratie, een techniek die bij uitstek geschikt is voor een domme computer die er niet tegenop ziet in snel tempo miljoenen optellingen en vermenigvuldigingen van dezelfde soort te maken. Daartoe schrijven we (14.2) eerst als een stelsel van twee differentiaalvergelijkingen van de eerste orde. Dat lijkt even of we het probleem ingewikkelder maken, maar schijn bedriegt. We vervangen de vergelijking (14.2) door

(14.3)
$$\mathrm{d}u/\mathrm{d}t = v$$
$$\mathrm{d}v/\mathrm{d}t = -\sigma^2 u - \mu v$$

De variabele v heeft daarbij de betekenis van snelheid. Om die vergelijkingen met behulp van de computer op te lossen vervangen we de differentiaalquotiënten door differentiequotiënten. Volgens de definitie is

$$\mathrm{d}u/\mathrm{d}t = \lim(u(t+h) - u(t))/h$$

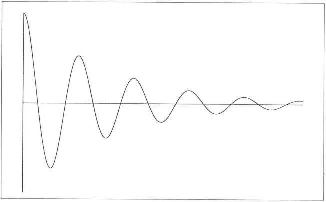
wanneer h, de zogenaamde stapgrootte, naar 0 gaat. Laten we de alleen theoretisch uitvoerbare limietovergang $h \rightarrow 0$ achterwege en nemen we voor h gewoon een klein getalletje, dan verwachten we normaal gesproken een goede benadering te krijgen. Voeren we die benadering in (14.3) zowel voor u als voor v uit, dan krijgen we

$$u(t+h) = u(t) + hv(t)$$
$$v(t+h) = v(t) - h(\sigma^2 u(t) + \mu v(t))$$

en dat kunnen we lezen als een iteratief dynamisch systeem, een modeltype dat in dit boek al eerder ter sprake is gekomen. We kunnen dat systeem dan ook in de volgende vorm brengen:

(14.4)
$$u_{n+1} = u_n + hv_n$$
$$v_{n+1} = -h\sigma^2 u_n + (1 - h\mu)v_n$$

Beginnen we met een gegeven startpositie $u = u_0$ en $v = v_0$, in fysische termen een gegeven uitwijking aan het begin en een gegeven beginsnelheid, dan kunnen we op de gebruikelijke wijze met de opvolgers u_n, v_n een baan vormen. De index n heeft nu de betekenis van het tijdstip nh gerekend vanaf het begin. In de bijbehorende grafische voorstelling hebben we kennelijk plaats u en snelheid v tegen elkaar uitgezet. We krijgen dan een zo-



figuur 14.1
Een gedempte trilling

genaamd *fasediagram*. Natuurlijk kunnen we ook de positie u tegen de tijd, of de index n, uitzetten, en dat is gebeurd in figuur 14.1.

Het bijbehorende programma, HARM, is heel eenvoudig, de elementaire tijdstap h is zo klein gekozen, dat de benaderingsmethode voldoende betrouwbaar is, terwijl met een moderne computer het programma toch in korte tijd voltooid kan worden. Hoewel men langs wiskundige weg veel theoretisch inzicht in de numerieke betrouwbaarheid van een dergelijke methode kan krijgen is in veel gevallen een experimentele aanpak wel zo effectief. Men voert eenvoudig hetzelfde programma uit voor een aantal waarden van h, te beginnen met een vrij grove waarde, om die dan bijvoorbeeld telkens te halveren. Bij elke verkleining gaat men dan na of er in de getoonde grafiek nog belangrijke veranderingen optreden. Zo niet, dan houdt men op.

De besproken benaderingsmethode, waaraan men vaak de naam van Euler verbindt, is universeel en algemeen bruikbaar. Voor wie heel nauwkeurig wil werken, bijvoorbeeld bij de berekening van de baan van een ruimtevaartuig, zijn betere technieken beschikbaar, zoals de naar Runge en Kutta genoemde methoden, maar voor eenvoudige simulaties is dat echt niet nodig.

In het speciale geval van de gedempte harmonische trilling zijn eenvoudige formules beschikbaar als

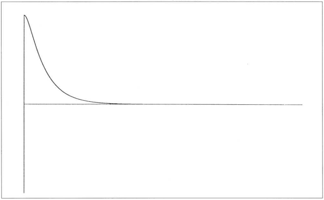
(14.5) $$u = ae^{-\mu t/2}\sin(\omega t + \alpha)$$

Door substitutie van deze formule in (14.2) vindt men na wat eenvoudig rekenwerk dat de frequentie ω gelijk is aan $\sqrt{(\sigma^2 - \mu^2/4)}$. Omdat uit een negatief getal geen reële wortel getrokken kan worden mag de dempingsconstante μ niet te groot zijn en moet aan de ongelijkheid

$$\mu < 2\sigma$$

zijn voldaan.

figuur 14.2
Aperiodieke demping



Wanneer de dempingsconstante zo groot is dat niet meer aan de ongelijkheid voldaan is, geldt de oplossing (14.5) niet meer en moeten we de volgende formule gebruiken:

$$(14.6) \qquad u = ae^{-pt} + be^{-qt}$$

waarbij p en q de wortels zijn van de vierkantsvergelijking

$$x^2 - \mu x + \sigma^2 = 0$$

De grafiek van u ziet er nu heel anders uit; geen herhaalde slingering meer, hooguit de aanzet van een slingering aan het begin. We spreken in dat geval van een *aperiodieke demping*. Die kan met hetzelfde programma worden verkregen, en we zien dan een grafiek als in figuur 14.2.

Relaxatie

Velen van ons kennen dat leuke demonstratiemodelletje dat nogal eens bij tentoonstellingen of in sommige tuinen onze aandacht moet trekken. Vanuit een kraan in een vaste positie stroomt water in de bakjes van een soort weegschaal. Wanneer een van de bakjes vol is slaat de weegschaal door, met het gevolg dat het andere bakje onder de kraan wordt geplaatst, terwijl het eerste bakje leegloopt. Wanneer het tweede bakje vol is slaat de weegschaal opnieuw door en komt het geleegde eerste bakje weer terug in de oorspronkelijke positie. Het proces herhaalt zich eindeloos zolang de kraan blijft stromen.

In wat meer fysische termen gevat kunnen we zeggen dat er door het lopen van de kraan energie aan het dynamische systeem wordt toegevoegd. Wanneer de opgenomen energie een bepaalde drempel overschrijdt, wordt de energie plotseling 'ontladen of losgelaten', een verschijnsel dat we *relaxatie* noemen. Het is een voorbeeld van een zoge-

naamde relaxatietrilling. In de natuur komen dergelijke relaxatietrillingen veel voor. Ze zijn in het algemeen gekenmerkt door een grote mate van stabiliteit, iets wat bij bijvoorbeeld biologische toepassingen zeer van belang is.

De Nederlandse wis- en natuurkundige Balthasar van der Pol bestudeerde omstreeks 1920 bij een elektrische schakeling een relaxatieverschijnsel met een trillingsvorm die we in figuur 14.3 afbeelden.

Daarbij slaagde hij erin een eenvoudig mathematisch model te ontwikkelen in de vorm van de volgende vergelijking:

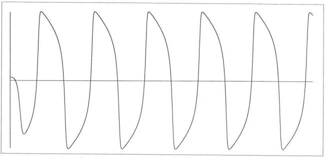
(14.7) $$d^2u/dt^2 + \mu(u^2 - 1)du/dt + u = 0$$

waarbij μ de zogenaamde *relaxatieparameter* is.

Deze vergelijking zou de geschiedenis ingaan als de vergelijking van Van der Pol. Over die vergelijking zijn talloze artikelen en boeken geschreven, en het is het begin geworden van een geheel nieuw hoofdstuk van wiskunde en mechanica.

Om die vergelijking enigszins te begrijpen vergelijken we hem met (14.1) en (14.2). Voor $\mu = 0$ is de vergelijking van Van der Pol identiek met die van de ongedempte harmonische trilling. We zouden dus verwachten dat voor een kleine waarde van μ de oplossingen van (14.7) maar weinig verschillen van een harmonische trilling, dat wil zeggen $u = a \sin(t)$ zou een goede benaderde oplossing zijn van de vergelijking van Van der Pol. Maar nu komt er iets uiterst belangrijks, onverwacht en nieuw voor wie voor het eerst met dit gebied kennis maakt.

Door de vergelijking (14.1) van de harmonische beweging is van de trilling wel de frequentie vastgelegd, maar niet de amplitude. Dat wil zeggen, voor elke waarde van a is $u = a \sin(\sigma t)$ een oplossing van de vergelijking. Bij de vergelijking van Van der Pol is dat niet meer zo. Is $u(t)$ daarvan een of andere oplossing, dan is bijvoorbeeld $2u(t)$ – of algemener $au(t)$ – geen oplossing meer, zoals een substitutie meteen laat zien. Dat komt doordat de variabele u in de vergelijking (14.1) op lineaire wijze voorkomt. In de vergelijking van Van der Pol staat naast een enkele u ook een term met u^2, en dat maakt heel veel uit. De vergelijking is niet meer li-



 figuur 14.3 Een relaxatietrilling voor $\mu = 3$

neair, en dergelijke vergelijkingen komen in de moderne wiskunde zoveel voor dat men ze *niet-lineaire vergelijkingen* is gaan noemen, en ze een plaats hebben gevonden in een geheel nieuw en belangrijk hoofdstuk van de wiskunde en de mechanica, *niet-lineaire wiskunde en mechanica*. In dat vakgebied worden allerlei dynamische systemen bestudeerd, en daarbij maakt men dankbaar gebruik van alles wat computers kunnen bieden.

Een kleine storing met grote gevolgen

Wanneer in de vergelijking van Van der Pol de relaxatieparameter erg klein is, dan wijkt de vergelijking maar heel weinig af van de harmonische vergelijking. Men mag dus wel veronderstellen dat een harmonische trillingsvorm als $u = a \sin(t)$ een goede benaderde oplossing is. Maar welke amplitude moeten we kiezen? Het verrassende is dat de vergelijking van Van der Pol een zeer bepaalde waarde van a selecteert, en wel $a = 2$. Ook al is de waarde van de relaxatieparameter μ nog zo klein, de amplitude van de benaderde harmonische oplossing is gefixeerd op de waarde 2. Nu weten we dat de oplossing van een differentiaalvergelijking sterk afhangt van de beginvoorwaarden, hier dus de waarden van u en du/dt op het begintijdstip $t = 0$. Maar hoe we ook beginnen (natuurlijk niet $u = 0$ en tegelijk $du/dt = 0$), steeds lijkt de oplossing na een aantal slingeringen op de harmonische trilling met amplitude 2. Het enige wat kan optreden is een faseverschil in de tijd, maar dat is onbelangrijk, slechts een kwestie van bijstellen van de tijdschaal. Er stelt zich met andere woorden na een zekere aanlooptijd een stabiele trilling in met een vaste amplitude.

Dat er altijd een trilling met een door het systeem bepaalde amplitude bestaat, kan theoretisch op de volgende wijze worden verklaard. De vergelijking van Van der Pol kunnen we opvatten als een voortdurend transport van energie. Vergelijken we (14.7) met (14.2), dan kunnen we de term $-\mu u(1-u^2)$ interpreteren als een wrijvingsterm μu van een gedempte harmonische slingering (men lette op het min-teken). Dat is eigenlijk alleen maar waar wanneer beide termen hetzelfde teken hebben, dat wil zeggen wanneer u in absolute zin groter dan 1 is. Wanneer daarentegen u juist klein is, dus $|u| < 1$, dan kloppen de tekens niet. We zouden dat moeten vergelijken met een harmonische slingering waarbij de demping negatief is. Dat betekent eigenlijk dat er tijdens de beweging energie wordt toegevoegd, dat dus de slingering sterker wordt en dat dientengevolge de maximale uitwijking steeds groter wordt. Blijkbaar is er een stabiele, zichzelf reproducerende beweging mogelijk, bij benadering een harmonische trilling met amplitude 2, waarbij door het systeem afwisselend energie opgenomen en weer afgestaan wordt. Gemiddeld verandert de totale energie daarbij niet.

Computerexperimenten

De vergelijking van Van der Pol kan niet met behulp van formules worden opgelost. Op zijn best kunnen de oplossingen worden voorgesteld door benaderende reeksen, maar daar moet theoretisch heel wat werk aan

verricht worden, en voor praktische doeleinden zijn ze van beperkte waarde. In zijn originele publikatie getroostte Van der Pol zich veel moeite om de vergelijking grafisch op te lossen, maar nu – met onze computer – is het werkelijk een fluitje van een cent, en dat laten we zien. Net als bij de vergelijking van de harmonische trilling vervangen we (14.7) door een stelsel vergelijkingen van de eerste orde:

(14.8) $\mathrm{d}u/\mathrm{d}t = v, \qquad \mathrm{d}v/\mathrm{d}t = -u + \mu(1 - u^2)$

In het programma POL1 laten we zien hoe daarmee voor een willekeurige waarde van μ de trillingsvorm verkregen kan worden. Het programma verloopt geheel volgens het schema van het programma HARM. De trilling wordt als functie van de tijd gestart met een kleine waarde van u, de uitwijking, terwijl de beginsnelheid nul is. De relaxatieconstante kan door de gebruiker worden gekozen. Beginnen we met een kleine waarde als μ = 0.1, dan zien we een aanloopperiode van een soort opslingering waarbij de amplitude toeneemt tot een waarde van ongeveer 2 zoals door de theorie voorspeld is of bevestigd kan worden. De uiteindelijke trilling is op het oog niet te onderscheiden van een harmonische trilling. Herhalen we het experiment met een wat grotere waarde van μ, bijvoorbeeld 0.8, dan zien we dat de aanloop wat korter is en dat de trilling sneller naar de limiettoestand toegaat. Bovendien valt het wellicht op dat een individuele golfbeweging niet meer symmetrisch is als bij een zuivere sinusvorm. Doen we het nog eens voor μ = 3.2 dan zien we een echte relaxatietrilling zoals afgebeeld in figuur 14.3. De fysische interpretatie van de asymmetrie is dat het oscillerende systeem telkens op een langzame wijze energie opneemt en die vervolgens snel weer afstaat (het systeem ontspant zich, relaxeert). Wie het programma gebruikt kan eventueel ook wijzigingen in de waarde van *t*max of in die van h aanbrengen. Een grotere tijdstap h betekent een verlies van nauwkeurigheid dat niet noodzakelijk zichtbaar behoeft te zijn, maar belangrijker is wel dat verdubbeling van h het programma tweemaal zo snel doet uitvoeren.

Fasediagram

In het programma POL2 worden positie u en snelheid v tegen elkaar uitgezet. Wanneer μ = 0, dan is de vergelijking van Van der Pol identiek met die van de ongedempte harmonische slinger. In plaats van (14.8) krijgen we eenvoudiger

$$\mathrm{d}u/\mathrm{d}t = v, \qquad \mathrm{d}v/\mathrm{d}t = -u$$

Hieruit volgt

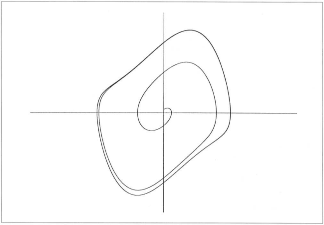
$$u\mathrm{d}u/\mathrm{d}t + v\mathrm{d}v/\mathrm{d}t = 0$$

ofwel

$$\mathrm{d}(u^2 + v^2)/\mathrm{d}t = 0$$

Dat betekent dat gedurende de beweging u^2+v^2 constant blijft. Meetkundig betekent dit dat in het u,v-vlak de punten (u,v) zich op een cirkel bevinden. Fysisch wil dat zeggen dat de som van potentiële en kinetische energie constant is. Kortom, de harmonische beweging manifesteert zich in het u,v-vlak als een cirkelvormige baan. In elke trillingsperiode wordt de baan een keer doorlopen.

Nu beschouwen we de algemenere situatie waarbij μ een zekere positieve waarde heeft. De theorie laat ons enigszins in de steek. Maar we kunnen wel voorspellen dat voor een kleine waarde van μ de te verwachten baan steeds meer op een cirkel zal gaan lijken, en dat die cirkel ongeveer de straal 2 zal hebben. Uitvoering van het experiment met het programma POL2 bevestigt dit vermoeden. Beginnen we met de startwaarden $u = 0.1$, $v = 0$, dan zien we dat voor bijvoorbeeld $\mu = 0.2$ de baan inderdaad naar een cirkel met een straal van ongeveer 2 toe spiraliseert. Wanneer echter μ wat groter is, zeg $\mu = 0.4$, dan ziet het er anders uit. De limietcirkel is gedeformeerd maar blijft een gesloten kromme. We spreken nu van een *limietcyclus* en die is weer het beeld in het fasevlak van de periodieke oscillerende beweging die op den duur ontstaat. Dat is overigens meer een theoretische uitspraak, want op het scherm zien we dat de limietcyclus al na een enkele slingering goed benaderd wordt. Het geval $\mu = 1$ laat het nog duidelijker zien. Hogere waarden van μ zijn ook mogelijk, maar dan moeten we de schaling in de WINDOW-opdracht aanpassen. De situatie voor $\mu = 0.8$ is afgebeeld in figuur 14.4.



figuur 14.4 De baan in het fasevlak voor $\mu = 0.8$

Een ouderwets horloge

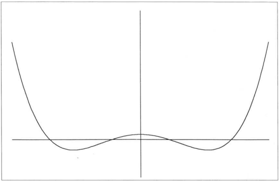
Het door Van der Pol opgestelde model kan op allerlei wijzen gevarieerd en uitgebreid worden. Met een kleine aanvulling kan men er een model van maken dat beschrijft hoe een stilstaand horloge, dat voor zijn energie nog gebruik maakt van een opgewonden veer, in gang kan worden gezet. Stilstand is hier een mechanisch evenwicht, en dat evenwicht is ook stabiel, omdat een klein tikje het horloge nog niet doet lopen. Een grote tik of wat schudden doet dat wel, en na korte tijd stelt zich een stabiele harmonische beweging met een vaste amplitude in als de limietcyclus van een Van der Pol-vergelijking. Als model, een generalisatie van (14.7), nemen we

$$d^2u/dt^2 + f(u)du/dt + u = 0$$

De nog nader te bepalen term $f(u)$ heeft de fysische betekenis van een demping. Wanneer de absolute waarde van u erg klein is, moet $f(u)$ positief zijn omdat het evenwicht $u = 0$ stabiel is. Ook voor grote waarden van $|u|$ moet u positief zijn. In het tussengebied, waar het systeem energie opneemt, moet $f(u)$ negatief zijn. Omdat het teken van u geen invloed heeft zorgen we ervoor – evenals in de vergelijking van Van der Pol – dat f van u^2 afhangt. Een mogelijke vorm, waarvan de grafiek in figuur 14.5 afgebeeld is, is

$$f = 0.2(u^2 - 0.2)(u^2 - 2)$$

Inderdaad voldoet volgens het programma POL3 het corresponderende model aan alle eisen. Bij de computerexperimenten brengen we het systeem uit balans door u aan het begin een voldoende grote waarde te geven. Daarbij kiezen we $v = 0$. Er zijn natuurlijk ook andere mogelijkheden, maar we laten het als gebruikelijk aan de lezer over in dat programma zelf kleine variaties aan te brengen. Men kan $f(u)$ een beetje anders kiezen of men kan als beginsituatie $u = 0$, met voor v een zekere waarde, nemen.



figuur 14.5
Grafiek van $f(u)$

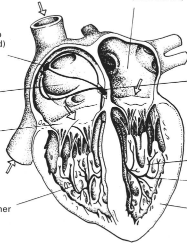
15 Het hart als elektrisch systeem

Het menselijk hart is een wonderbaarlijk orgaan dat aan de basis van ons leven staat. Al in een heel vroeg stadium, ver voor de geboorte, kan men in een zich ontwikkelend organisme een kloppend hart waarnemen terwijl andere details zich nog aan waarneming onttrekken. Zodra het hart ophoudt met kloppen is er gevaar voor het leven, maar anderzijds kan een hart ook na de dood van de eigenaar nog voldoende levenskracht bezitten om elders als 'ruilhart' te existeren. Met een hart kan veel misgaan, maar de medische wetenschap beschikt tegenwoordig over vele mogelijkheden om een gebrekkig functionerend hart tot betere prestaties te brengen. Een van die mogelijkheden is het aanbrengen van een *pacemaker* waarbij het hart extern elektrische prikkels krijgt toegediend wanneer de eigen voorziening het laat afweten. In dit hoofdstuk proberen we aan de hand van een aantal simulaties inzicht te verwerven in de werking van het hart en van een pacemaker.

De elektrische eigenschappen van het hart

Het hart, schematisch afgebeeld in figuur 15.1, kunnen we opvatten als een vloeistofpomp die bestuurd wordt door elektrische prikkels. Alles wat te maken heeft met stromingsproblemen laten we buiten beschouwing, ons interesseert alleen de 'elektrische bedrading'. De elektrische motor van alles is de zogenaamde *sinusknoop*, een soort natuurlijke pacemaker. De sinusknoop genereert periodiek een elektrisch signaal, een aantal prikkels, waardoor allereerst de boezems zich samentrekken. De prikkels bereiken vervolgens een tweede zenuwknoop, de *AV-knoop* (en de zogenaamde bundel van His), een tweede elektrische motor die op zijn beurt zorgt voor het samentrekken van de hartkamers. In beginsel zijn er dus twee biologische oscillatoren actief waarbij de tweede door de eerste wordt beïnvloed.

Wanneer alles goed gaat lopen beide oscillatoren met elkaar in de pas. Wanneer de eerste oscillator een volledige cyclus heeft doorlopen, heeft ook de tweede zijn kringloop volbracht. We kunnen denken aan het beeld van twee onderling gekoppelde klokken die even snel lopen. Maar met klokken, en ook met het hart, kan wel eens wat misgaan. Stel eens dat de tweede klok het tempo van de eerste niet kan bijhouden en maar drie omlopen voltooit wanneer de eerste klok vier keer rond gaat. De snelheden van de klokken verhouden zich dus als 4 : 3. Er bestaat dan toch een herhaalbare situatie, een soort koppeling van de twee uurwerken in meer algemene zin. In de technische literatuur spreekt men dan van een *fasekoppeling* ter grootte van 4 : 3. Over het begrip fase zeggen we aanstonds



figuur 15.1 Het hart als elektrisch systeem

figuur 15.2 Het fenomeen van Wenckebach

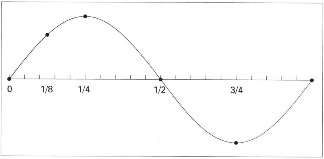
meer. Bij het hart kan er een ritmestoornis zijn als geschetst in figuur 15.2.

De oscillator van de AV-knoop reageert dan wat traag op de ontvangen prikkels, en het duurt telkens wat langer voordat de kamers zich na de boezems samentrekken. Zoals in het schetsje te zien is komt het een keer voor dat op de samentrekking van de boezems geen samentrekking van de kamers volgt, het hart slaat even over zouden we kunnen zeggen. Dit verschijnsel staat bekend als het fenomeen van Wenckebach, een beroemde Nederlandse cardioloog (geboren in Den Haag in 1864, gestorven in Wenen in 1940) die daaraan veel aandacht heeft besteed. Medici kennen nog veel meer van zulke afwijkingen, maar liever keren we terug tot de wiskunde.

Faserekenen

Het begrip fase komt met verschillende betekenissen in de natuurwetenschappen voor. Het is afgeleid uit het Grieks als een aanduiding voor een verschijnsel, meestal iets van periodieke aard zoals de schijngestalten van

figuur 15.3 Fase bij een sinusgolf



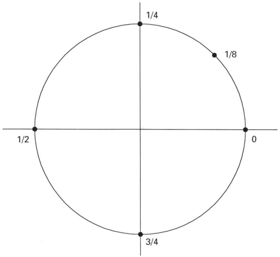
de maan. Volle maan zou men in getallenmaat als fase 0 kunnen aanduiden met de nieuwe maan als fase 1/2. Tijdens zijn cyclus groeit de fase van de maan dan aan van de fasegrootte 0 tot aan 1 bij de volgende volle maan, een soort klok die telkens bij het bereiken van de waarde 1 naar 0 terugspringt. Wiskundig heet dat rekenen modulo 1. In de natuurkunde is het begrip fase vooral nuttig bij harmonische trillingen, bij sinusvormige bewegingen zoals die bijvoorbeeld optreden bij licht en geluid. In figuur 15.3 hebben we het het begrip fase toegelicht aan de hand van een enkele sinusgolf.

Hier hebben we meer met klokken te maken en dan kunnen we de stand van een klok, de fase dus, beter weergeven als een (lopend) punt op een cirkel, net als bij een echte klok. In de wiskunde wordt de draaizin positief gerekend tegen die van een uurwerk in, tegenwijzerzin. We moeten daar even op letten wanneer we wiskundige klokken met echte klokken vergelijken. De fase van een klok is trouwens ook een zaak van afspraak. Vaak kiest men de stand van drie uur als die van fase 0, maar soms wordt twaalf uur als fase 0 opgevat. Erg belangrijk is dat niet, want alleen faseverschillen hebben echte betekenis. Een wiskundige klok kunnen we modelleren als

$$x = \cos(2\pi t/T), \qquad y = \sin(2\pi t/T)$$

Daarbij heeft t de betekenis van tijd en T is de omlooptijd. De grafiek van de punten (x,y) is een cirkel met straal 1. Elk punt van de cirkel karakteriseert een toestand van de klok, een fase. Interpreteren we de beginpositie $t = 0$ als fase 0 – het punt P met de coördinaten (1,0) staat dan in de positie van drie uur – dan correspondeert een willekeurige positie op het tijdstip t met de fase t/T mod 1. In plaats van T kunnen we ook de omloopsnelheid $v = 1/T$ invoeren. In figuur 15.4 is de fase op de door x,y bepaalde cirkel aangegeven.

In een aantal computersimulaties moet op herhaalde wijze met klokken gerekend worden. Daarbij moeten telkens fasewaarden bij elkaar worden opgeteld of van elkaar afgetrokken. In principe moet elke waarde steeds worden gereduceerd tot een waarde tussen 0 en 1, maar soms is het handig om tijdens het rekenproces die reductie tot een later moment uit te



figuur 15.4 Fase bij een wiskundige cirkelklok

stellen. Dat zelfde is ook in de goniometrie gebruikelijk wanneer we hoeken (in graden bijvoorbeeld) bij elkaar willen optellen. We schrijven dan in eerste instantie iets als

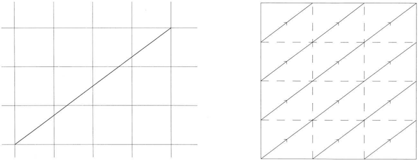
$$160^\circ + 220^\circ + 90^\circ + 150^\circ + 300^\circ = 920^\circ$$

Na aftrekken van 2 ⋆ 360° moeten we de uitkomst van de optelling als een hoek van 200° interpreteren.

Periodiek ruitjespapier

Modellen waarbij met fasegrootheden wordt gerekend kunnen grafisch op uit vierkanten opgebouwd grafiekenpapier worden weergegeven waarbij elk vierkant identiek is met elk ander vierkant. In figuur 15.5 hebben we twee diagrammen getekend. Links is op de gewone wijze (op ruitjespapier) een grafiek getekend van de rechte lijn $y = 0.75x$ waarbij $0 < x < 4$ en $0 < y < 3$. Maar wanneer x en y de betekenis van fases hebben is elk punt (x,y) equivalent met (x_0,y_0) waarbij x_0 en y_0 de modulo 1 gereduceerde waarden van x en y zijn. Noemen we het vierkant met de hoekpunten (0,0) en (1,1) het hoofdvierkant, dan wil dat zeggen dat elk punt van de grafiek, hier dus de rechte lijn, dat buiten het hoofdvierkant ligt teruggebracht moet worden tot een equivalent punt binnen het hoofdvierkant. In het rechterdiagram van figuur 15.5 is die reductie uitgevoerd voor de rechte lijn.

Het hoofdvierkant geeft in principe alle informatie die nodig is. Zoals we in de figuur zien verdwijnt soms een stuk van de rechte lijn aan een zij-



figuur 15.5 Gewoon en periodiek ruitjespapier

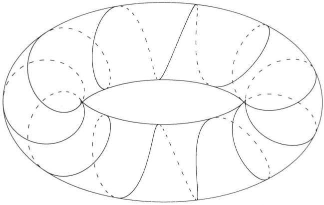
de van het hoofdvierkant om zich aan de andere zijde in een overeenkomstige positie voort te zetten. Omdat fase 1 hetzelfde is als fase 0 zijn punten $(1,y)$ op de rechterzijde eigenlijk identiek met punten $(0,y)$. Hetzelfde geldt voor de punten $(x,1)$ en $(x,0)$ van boven- en onderrand.

Wie een goed ruimtelijk inzicht heeft kan het volgende gedachtenexperiment uitvoeren. Hij neemt als hoofdvierkant een vierkant vel papier en buigt dit in de verticale richting rond tot een horizontale cilindrische koker. Boven- en onderrand van het vierkant worden daarbij aan elkaar vastgeplakt. De koker wordt nu in horizontale zin rondgebogen – het papier rekt of krimpt daarbij wat mee – en de twee cirkelvormige uiteinden worden aan elkaar vastgemaakt. Wat dan ontstaat heet een *torus*, een object dat doet denken aan een werpring of een binnenband. Amerikanen die meer vertrouwd zijn met produkten als hamburgers en doughnuts worden altijd getroffen door de gelijkenis van het meetkundige lichaam met hun populaire lekkernij. Overigens schijnt ook hier de 'donut' ingeburgerd te raken. We zouden dan aan figuur 15.5 nog een grafische voorstelling kunnen toevoegen waarbij de rechte lijn zich als een kronkellijn over het torusoppervlak beweegt.

Uiteraard is tekenen op een plat vierkant heel wat eenvoudiger dan tekenen op een torus, gevolgd door een projectie op een plat vlak. Maar toch kan dat torustekenen, denkbeeldig of desnoods schetsmatig, ons helpen inzicht te krijgen in fasemodellen.

Torustekenen

Het programma TORUS stelt de lezer in staat om op het beeldscherm de schijnbare omtrek van een torusoppervlak te verkrijgen en daarover een willekeurige 'rechte' lijn te laten kronkelen. Van de wiskundige achtergrond vermelden we alleen dat het torusoppervlak in ruimtelijke coördinaten x,y,z bepaald is door



figuur 15.6
Een kronkellijn op een torus met fasekoppeling

$$x = (a + b \cos t) \cos u$$
$$y = (a + b \cos t) \sin u$$
$$z = b \sin u$$

De coördinaten van een punt op de torus zijn gegeven door t en u waarbij $t/(2\pi)$ en $u/(2\pi)$ de betekenis van fase hebben en daarmee overeenstemmen met de horizontale coördinaat x en de verticale coördinaat van het vlakke periodieke vierkant. De rechte lijn van figuur 15.5 zou dan op de torus door $u = 0.75t$ bepaald kunnen zijn.

Het gaat eigenlijk nog steeds om een grafische illustratie van het fenomeen van Wenckebach, de fasekoppeling van 4 : 3, waarbij de horizontale coördinaat x van het periodieke vierkant en t van het torusoppervlak viermaal rondgaat terwijl de verticale coördinaat y of u drie keer rondloopt. In plaats van 4 : 3 kunnen we elke andere verhouding nemen, bijvoorbeeld $p : q$ waarbij p en q gehele getallen zijn en p/q een onvereenvoudigbare breuk is. Op het torusoppervlak komt dan een gesloten kronkellijn die horizontaal p-maal en verticaal q-maal rondloopt. Wanneer de verhouding irrationaal is sluit de kronkellijn zich nooit en vormt een soort dichte overdekking van het torusoppervlak, een beeld dat de gedachte aan chaos oproept. In figuur 15.6 is om illustratieve redenen gekozen voor een fasekoppeling van 1 : 11.

Een treurig verhaal

In 1914 was een jong fysioloog aan de McGill University School of Medicine van Montreal bezig met experimenten waarbij hij onderzocht hoe een hart reageert op kleine elektrische prikkels. Hij had een apparaat gebouwd waarmee hij kleine stroomstootjes kon opwekken met een nauwkeurig regelbare sterkte en frequentie. Experimenten met proefdieren waren al achter de rug en nu zou hij voor het eerst een mens als proef-

object gaan nemen. Als consciëntieus wetenschappelijk onderzoeker begon hij met zichzelf, en hij had er het weekend van zaterdag 7 november helemaal voor vrijgehouden...

Maandagmorgen doet in alle vroegte de conciërge zijn ronde. Te midden van een chaos aan elektrische apparaten en draden ligt Mines uitgestrekt op de vloer. Een van de apparaten registreert een fibrillerend hart, dat van Mines. Mines heeft zijn experiment niet overleefd.

Het leek lange tijd onbegrijpelijk dat een zwak stroomstootje in staat was een hart grondig te ontregelen, maar nu kunnen we het verschijnsel verklaren aan de hand van een eenvoudig wiskundig model. Van der Pol heeft er ons bewust van gemaakt dat stabiele periodieke verschijnselen in feite limietcycli zijn van dynamische systemen. In het hoofdstuk over de relaxatietrillingen hebben we gezien dat bij de vergelijking van Van der Pol de limietoplossing, een periodieke trilling, niet afhangt van de begintoestand. Dat betekent dat in dat model een externe verandering, in mechanische zin een verstoring van de evenwichtsstand, het systeem steeds weer tot dezelfde stabiele trilling terugvoert.

Voor de beschrijving van een stabiel kloppend hart kan het model van Van der Pol technisch vereenvoudigd worden tot

$$\mathrm{d}r/\mathrm{d}t = \mu r(1 - r^2)$$
$$\mathrm{d}\varphi/\mathrm{d}t = 2\pi$$

waarbij r en φ poolcoördinaten zijn. De evenwichtsoplossing is $r = 1$ waarbij $\varphi = 2\pi t$, zodat t mod 1 als de fase opgevat kan worden. In x,y-coördinaten is deze oplossing hetzelfde als

$$x = \cos(2\pi t), \qquad y = \sin(2\pi t)$$

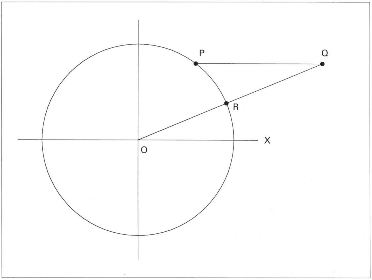
een gewone cirkelklok. Maar er is meer. Wanneer de bovenstaande vergelijkingen geïntegreerd worden – een eenvoudige zaak – kan de oplossing worden geschreven in de vorm

$$1/r^2 = 1 + C\exp(-\mu t), \qquad \varphi = 2\pi t + \text{constante}$$

De constante in de uitdrukking voor φ heeft alleen met de nulpositie van de klok te maken. Stellen we de klok in op nul aan het begin van het experiment, dan kunnen we die constante gewoon vergeten. De constante C in de uitdrukking voor r is daarentegen heel erg belangrijk. Gaan we uit van een beginpositie waarbij r niet gelijk aan 1 is, dan kunnen we C zodanig aanpassen dat r de voorgeschreven beginwaarde heeft voor het begintijdstip $t = 0$. Is bijvoorbeeld $r = 2$ voor $t = 0$ dan moet C gelijk zijn aan -3/4. Maar gaat de klok lopen, dan neemt de bijdrage van $\exp(-\mu t)$ af naarmate t toeneemt. Ten slotte is de waarde van die exponentiële functie praktisch niet meer van nul te onderscheiden. In dat geval is de waarde van r praktisch gelijk aan 1. Grafisch ziet de baan van het punt (r,φ) eruit als een spiraal die begint op een afstand 2 van de oorsprong, maar die snel tot de eenheidscirkel, de limietcyclus in de zin van Van der Pol, nadert. Naarmate de waarde van de factor μ groter is gaat dat sneller.

In figuur 15.7 hebben we het nog eens grafisch verduidelijkt. Elk punt op de eenheidscirkel stelt een stabiele toestand van de hartklok voor. In het punt P heeft het hart een bepaalde fase f als bepaald door de poolhoek POX. Een storing brengt P uit de evenwichtsstand ergens buiten de cirkel naar het punt Q. We nemen in deze simulatie aan dat een storing altijd neerkomt op een horizontale verplaatsing naar rechts. De lengte a van PQ is een maat voor de intensiteit van de storing. Een waarde van a in de buurt van 1 betekent een zwakke storing, zoals in het experiment van Mines. Na de storing keert de klok weer terug naar een positie op de eenheidscirkel. Is μ vrij groot dan gaat dit als het ware momentaan en is het punt R, het snijpunt van OQ met de eenheidscirkel, de nieuwe stabiele positie. De hartklok loopt dan weer gewoon verder volgens punten op de eenheidscirkel. De illustratie lijkt ons duidelijk te maken dat een storing van willekeurige grootte van een willekeurige stabiele situatie niet veel anders lijkt te zijn dan een momentane faseverandering. In de figuur wordt de fase van P teruggezet naar die van Q.

Er is echter een zeldzame uitzondering. Wanneer de fase 1/2 is, de positie $P_1(-1,0)$, en de intensiteit a precies gelijk is aan 1, dan komt het punt Q terecht in de oorsprong O. De lijn OQ is dan onbepaald, evenals het punt R. We kunnen ook zeggen dat door de storing het beeldpunt Q van P terechtgekomen is in de oorsprong O, een instabiel evenwicht van het klokmodel, een punt ook waar het begrip fase geen zin meer heeft. Een



figuur 15.7 Een storing van de hartklok

overeenkomstige wiskundige redenering zegt dat in het bovenstaande klokmodel elk punt kan dienen als beginpunt van een tot de eenheidscirkel spiraliserende baan van de oorsprong.

In het programma KLOK laten we zien hoe de hartklok reageert op een willekeurige storing. De waarde van μ kan willekeurig gekozen worden. Aanbevolen wordt een paar waarden van μ te proberen, te beginnen met bijvoorbeeld $\mu = 0.1$, om te eindigen met bijvoorbeeld $\mu = 2$. De storing vindt in dit model steeds plaats op de meest ongunstige positie van fase 1/2. De intensiteit a van de stimulus kan weer willekeurig gekozen worden. Wetende dat $a = 1$ een fatale afloop symboliseert, zou men behalve met waarden als $a = 0.2$ en $a = 3$ ook eens iets als $a = 0.99$ of $a = 1.02$ kunnen proberen.

In het interactieve programma KLOKM blijft de hartklok normaliter op stabiele wijze doorlopen. Met een willekeurige toetsaanslag (de spatiebalk bijvoorbeeld) wordt een stimulus van de grootte a gegeven. Op het scherm ziet men hoe de klok hierop reageert en hoe het wiskundige hart weer opnieuw tot stabiele actie komt. De factor μ kan ook in dit programma vooraf willekeurig gekozen worden. Het experiment kan willekeurig worden herhaald totdat de stimulus in de gevaarlijke faseperiode wordt toegediend of men het programma gewoon wil beëindigen.

Het pacemaker-model

Tot zijn eenvoudigste vorm teruggebracht geeft figuur 15.7 alles wat nodig is om te begrijpen wat de reactie van een hart op een externe stimulus is. Is f de fase van het punt P, dan vindt er ten gevolge van een stimulus van de grootte a een faseverandering $f \to F(f,a)$ plaats. De werking van een pacemaker berust op het periodiek toedienen van een externe stimulus. We stellen ons voor dat telkens op het moment dat de stimulus wordt toegediend een meting wordt verricht. Op het moment van de stimulus is de fase van de hartklok gelijk aan f, na de stimulus is de fase veranderd in $F(f,a)$. De faseklok loopt nu door met een vast interval c totdat opnieuw een stimulus wordt toegediend. Na deze cyclus is de fase veranderd volgens het schema

$$f \to c + F(f,a)$$

De functie F kan met wat goniometrie worden berekend door in figuur 15.7 de hoek QOX in verband te brengen met de hoek POX, maar de berekening laten we in deze tekst achterwege. We gaan liever verder met een eenvoudige benaderde uitdrukking van F die leidt tot de veel bestudeerde cirkelafbeelding geschreven als het iteratieve proces

$$x_{n+1} = x_n + c - a\sin(2\pi x_n)/(2\pi) \quad \mod 1$$

Dat model vertoont in kwalitatieve zin alles wat we met een goed wiskundig model van de hartritmiek zouden kunnen verwachten. We herhalen nog even dat bij dit pacemaker-model a de grootte van de pacemaker-stimulus en c de periode van de stimulus is. De betekenis van x_n en van x_{n+1}

is die van de fase aan het begin en aan het eind van de cyclus. Het model is een fraai voorbeeld van een fase-fase-afbeelding die zich goed leent voor afbeelding op een torusoppervlak of, eenvoudiger, op een periodiek vierkant.

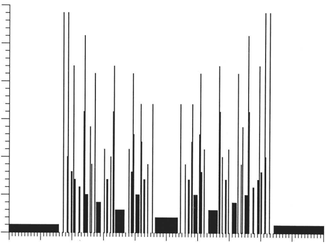
Simulaties

Met het pacemaker-model kunnen we verschillende computerexperimenten uitvoeren. Daartoe is een aantal programma's beschikbaar met de gemeenschappelijke titel CIRKELH.

In het eerste programma, CIRKELH1, wordt het gedrag van het iteratieve pacemaker-model beschouwd voor willekeurig te kiezen waarden van c en van a. De fase wordt in dit programma voorbij 1 doorgeteld. Eenvoudig is te zien dat er in sommige gevallen een aantrekkend dekpunt of een periodieke aantrekkende cyclus bestaat, fasekoppeling dus. In vele gevallen is er chaos, maar dat is in dit model niet goed waar te nemen.

In het tweede programma, CIRKELH2, wordt de fase telkens tot (0,1) gereduceerd en wordt in het iteratieproces alleen het hoofdvierkant weergegeven. Gevallen van fasekoppeling en van chaos zijn hierin duidelijker te zien.

In het programma CIRKELH4 kiezen we wederom a vast (bijvoorbeeld $a = 1$) en laten we c met kleine stapjes van 0 tot aan 1 lopen. Voor elke waarde van c waarvoor een stabiele periodieke q-cyclus bestaat met $q < 30$ wordt in het diagram een verticaal lijntje van de lengte q getekend. Duidelijk zien we hoe grillig de door q bepaalde fasekoppeling verloopt



figuur 15.8 Een diagram van fasekoppeling voor $a = 1$

als functie van c. Over het patroon zou nog veel meer te zeggen zijn, maar hier merken we slechts op dat het patroon ten opzichte van $c = 0.5$ symmetrisch is. Het diagram behorende bij de waarde $a = 1$ is afgebeeld in figuur 15.8.

Het programma CIRKELH5 geeft een fraai diagram van fasekoppeling waarin gebruik van kleuren wel noodzakelijk is. De rechthoek bepaald door de grenzen $0 < c < 1$ en $0 < a < 2\pi$ correspondeert op het beeldscherm met een rechthoek van pixels. Voor elke pixel, equivalent aan een punt (c,a), kan een periode q, de noemer van een eventuele fasekoppeling $p : q$, worden berekend. Vaak is q een niet te groot getal, maar wanneer dat getal hoog is en er dan een mogelijkheid van chaos bestaat, wordt $q = 0$ genomen. Volgens de kleurencode $1+(q \bmod 14)$ wordt hieruit een kleur van de pixel afgeleid. Wanneer het programma klaar is, na vele uren wellicht, is een fraai diagram verkregen dat is afgebeeld in figuur 15.9 (zie bladzijde 87 in het kleurkatern).

16 Orde en chaos

Over chaos is al veel geschreven, veel zinnigs maar ook veel onzin. Chaos lijkt een nieuw modewoord te zijn dat een wetenschappelijke verklaring zou moeten geven voor veel dingen die we niet begrijpen. Economen spreken bijvoorbeeld over 'de chaostheorie' alsof onvoorspelbare economische fluctuaties aan een theoretische wetmatigheid onderworpen zouden zijn, en op grond daarvan dus voorspelbaar en beheersbaar. Het probleem is dat begrippen als orde en chaos in wetenschappelijke zin geen precieze universele interpretatie toelaten, en dat ze in de omgangstaal vaak op meer intuïtieve wijze worden gebruikt. In dit hoofdstuk zullen we trachten aan de hand van een paar wiskundige modellen en simulaties enige orde in de chaos te brengen.

We stellen voorop dat orde in beginsel wijst op regelmaat, herhaalbaarheid en periodiciteit. Chaos is afwezigheid van orde. De ontwikkeling van een rationale breuk in het decimale talstelsel is, afgezien van eventueel een paar getallen aan het begin, een getallenrij die bestaat uit een zich telkenmale herhalend stukje. Zo is

$$20/7 = 2.857142\ 857142\ \ldots$$

en dat is dus een voorbeeld van orde.

De ontwikkeling van een onmeetbaar getal als π is niet periodiek

$$\pi = 3.141592\ 653589\ \ldots$$

en we zouden van chaos kunnen spreken. Toch staat elke decimaal in deze ontwikkeling rigoureus vast. Met 'wat rekenwerk' kunnen we zelfs zonder veel moeite een willekeurig rijtje decimalen, zeg de rangnummers 100000-101000, opschrijven. Veel vreugde zullen we daaraan niet beleven, want het rijtje onderscheidt zich nauwelijks van een overeenkomstige reeks van duizend op toevallige wijze gekozen getallen van 0 tot 9. Statistische toetsen kunnen dat bevestigen. Dat zelfde geldt voor 'eenvoudiger' onmeetbare getallen als $\sqrt{2}$ en $(-1+\sqrt{5})/2$, het gulden-snedegetal.

We beperken ons bij de discussie van orde en chaos tot eenvoudige dynamische systemen, modellen die met inachtneming van een aantal vereenvoudigingen en idealisaties als model kunnen dienen voor in de natuur optredende verschijnselen. Wie met de term 'dynamisch systeem' nog onvoldoende vertrouwd is kan daarbij het beste denken aan het voorbeeld van een mechanisch systeem dat onder invloed van krachten in beweging is, hemellichamen bijvoorbeeld. In uitgebreide zin is ook het hieronder besproken model van de geremde groei, het Verhulst-model, een dynamisch systeem.

De transformatie van de bakker

In de wetenschappelijke wereld is veel aandacht besteed aan een paar heel eenvoudige modellen waaraan op het gebied van chaos heel veel te beleven is. Dat zijn onder andere 'the baker's transformation, Arnold's cat, the Verhulst model'. Het algemene beeld bij deze modellen is dat er vele vormen van chaos bestaan en dat de mate van orde en chaos met een of meer karakteristieke parameters gemeten kan worden.

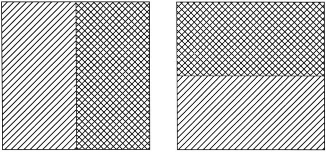
De *transformatie van de bakker* heeft betrekking op de transformatie van een vierkant $0 < x < 1$, $0 < y < 1$ waarbij de linkerhelft $0 < x < 1/2$ afgebeeld wordt op de onderhelft $0 < y < 1/2$. Analoog wordt de rechterhelft $1/2 < x < 1$ afgebeeld op de bovenhelft $1/2 < y < 1$. Het doet denken aan de bewerking van een vierkant blok deeg. Eerst wordt het blok geplet tot een rechthoek $0 < x < 2$, $0 < y < 1/2$. Daarna wordt het in twee helften gehakt en wordt de rechterhelft op de linkerhelft geplaatst als weergegeven in figuur 16.1.

De bakker voert deze transformatie enige malen achter elkaar uit met het doel een goede menging van het deeg te krijgen. In de wiskundige vorm van een iteratief model hebben we de beschrijving

$$
\begin{array}{llll}
x_{k+1} = 2x_k, & y_{k+1} = y_k/2 & \text{voor} & x_k < 0.5 \\
x_{k+1} = 2x_k - 1, & y_{k+1} = (1 + y_k)/2 & \text{voor} & x_k \geq 0.5
\end{array}
$$

De essentie is dat lijntjes in horizontale richting met een factor 2 uitgerekt en in verticale richting met dezelfde factor verkort worden. Wat er gebeurt met een klein rechthoekig blokje kunnen we op het beeldscherm vervolgen met een simulatie van dit model zoals in het programma BAKERB.

Wanneer we trachten met behulp van een zelfde simulatie de baan van een 'willekeurig' punt te vervolgen, lopen we de kans te botsen met de beperkingen van deze wereld waarin geen plaats is voor willekeurige onmeetbare getallen. De x-coördinaat van het startpunt is in zijn theoretische algemeenheid een onmeetbaar getal met oneindig veel decimalen. In de computerwereld werken we met het tweetallige stelsel, en dat betekent dat de beginwaarde van x een oneindig lange reeks van nullen en enen is.



figuur 16.1 De bakkerstransformatie

Omdat in deze transformatie bij elke iteratiestap x met de factor 2 vermenigvuldigd wordt heeft dat tot effect dat de hele rij binalen in de ontwikkeling van x een plaats naar links opschuift. De echte bakker houdt zeker bij tien stappen al op, hij heeft dan al bladerdeeg van 1024 plakjes gekregen! Maar wiskundigen willen veel verder, om de echte chaos (of orde) te zien. Zij zijn dus benieuwd naar de baan van het beginpunt wanneer het aantal stappen zeg honderd of nog veel groter is. Helaas beschikt de computer maar over een beperkt aantal binalen van de beginwaarde x_0 van x. Dat aantal kan iets in de orde van 20 of 40 zijn, maar veel maakt het niet uit. Na honderd iteratiestappen is er allang niets meer van over. Wat we registreren als de honderdste binaal van x_0 is dus pure fantasie, iets wat af kan hangen van de inwendige bouw van de computer of van de gebruikte programmeertaal, of gewoon iets wat door toeval gemaakt is. Soms vult de computer ontbrekende binalen met nullen aan en dan loopt de baan zeker vast in $x = 0$ (en $y = 0$).

Om aan dat soort problemen te ontkomen wijzigen we in een computerprogramma de bakkerstransformatie door

$$\text{if } x_k<0.5 \text{ then } y_{k+1}=y_k/2 \text{ else } y_{k+1}=(1+y_k)/2$$
$$x_{k+1} = 2x_k + a \text{ rnd mod } 1$$

waarbij a een heel klein getal als 2^{-20} is en rnd de bekende uitdrukking voor een toevallig getal tussen 0 en 1. Deze truc is uitgevoerd in het programma BAKERP. In dat programma wordt een baan weergegeven van tienduizend punten. Het resultaat is dat het vierkant gelijkmatig met punten wordt overdekt, een resultaat dat we mogen interpreteren als pure chaos, als een voorbeeld van goed mengen.

We nodigen de lezer uit dat programma uit te voeren met verschillende waarden van a, maar niet $a = 0$ want dan kan het mislopen.

Volledigheidshalve vermelden we de volgende uitbreiding van de bakkerstransformatie

$$x \to 2x, \quad y \to by/2 \quad \text{voor} \quad x < 0.5$$
$$x \to 2x - 1, \quad y \to (1 + b)y/2 \quad \text{voor} \quad x \geq 0.5$$

waarbij b een positief getal kleiner dan 1 is. Bij deze transformatie kunnen we denken aan deeg dat tijdens de behandeling in de verticale richting krimpt, zodat er telkens nieuwe horizontale laagjes aangevuld moeten worden om weer de vierkante vorm terug te krijgen.

De kat van Arnold

De Russische wiskundige Arnold is een van de belangrijkste moderne autoriteiten op het gebied van dynamische systemen, orde en chaos. Op Arnolds naam staan vele uitstekende boeken en artikelen. Als weinigen verstaat hij de kunst met eenvoudige voorbeelden moeilijke begrippen duidelijk te maken.

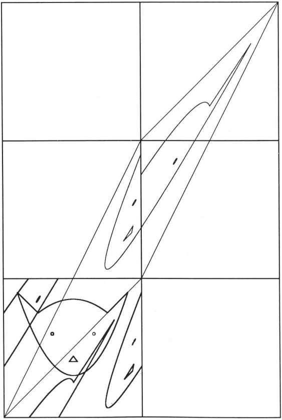
Een van zijn bekendste modellen betreft een conservatief iteratief dynamisch systeem waarbij een periodiek vierkant vervormd wordt tot een

parallellogram als geschetst in figuur 16.2. De toevoeging 'conservatief' is weer ontleend aan de mechanica en heeft te maken met de geldigheid van een behoudswet, de wet van behoud van energie bijvoorbeeld. Hier krijgt het in een iteratief model, een model waarin telkens een meetkundige transformatie wordt herhaald, de betekenis van het behoud van oppervlakte.

De wiskundige beschrijving is

$$x \rightarrow x+y \mod 1, \qquad y \rightarrow x+2y \mod 1$$

Wanneer we in de formules de toevoeging mod 1 even weglaten, dan gaat het hoofdvierkant met de hoekpunten (0,0),(1,0),(1,1),(0,1) over in het parallellogram met de hoekpunten (0,0),(1,1),(2,3),(1,2). Dat parallellogram heeft dezelfde oppervlakte als het oorspronkelijke vierkant. Maar



figuur 16.2
Arnolds kat

eigenlijk moeten we de transformatie van Arnold opvatten als een transformatie op een torusoppervlak, zoals we die in het vorige hoofdstuk hebben besproken. Het hoofdvierkant staat dan model voor het gehele torusoppervlak. De mod 1 toevoeging betekent dat alles wat door de transformatie buiten het hoofdvierkant terechtkomt daarin weer wordt teruggebracht. We werken in feite op periodiek ruitjespapier als besproken in het vorige hoofdstuk.

Arnold laat zien hoe – zoals in figuur 16.2 – de kop van een kat, opgebouwd uit een paar eenvoudige meetkundige lijntjes, vervormd en verknipt wordt. Oorspronkelijk bevindt de kattekop zich in het hoofdvierkant. Door de transformatie gaat het hoofdvierkant over in een parallellogram en komen delen van de kat in andere vierkanten terecht. Die delen moeten naar het hoofdvierkant worden teruggebracht. Dat betekent meetkundig dat het parallellogram in vier stukken wordt geknipt en dat de vier delen als een legpuzzel op de plaats van het hoofdvierkant aan elkaar worden gevoegd. Uiteraard is daarbij de kattekop in vreemde stukken uiteengevallen.

Wanneer de kat van Arnold op iteratieve wijze enige malen behandeld is lijkt het wel of alle onderdelen op chaotische wijze her en der verspreid zijn geraakt. Dat is ook de bedoeling van Arnold: laten zien hoe bij een heel eenvoudig model reeds chaos en volledige menging kan ontstaan. Wanneer de transformatie van Arnold maar vaak genoeg wordt herhaald bevat elk stukje van het hoofdvierkant, hoe klein ook, wel onderdeeltjes van de versnipperde kat. Dat zelfde geldt natuurlijk ook bij de interpretatie van de kat op het torusoppervlak.

Zowel bij de bakkerstransformatie als bij de 'cat map' is het dynamisch systeem chaotisch in de volgende betekenis. Twee banen die tegelijk beginnen in twee naburige punten wijken uit elkaar en gaan steeds meer een eigen leven leiden, zonder iets met elkaar te maken willen hebben, een uitdrukkingswijze die desgewenst vervangen mag worden door de constatering dat de onderlinge correlatie verdwijnt.

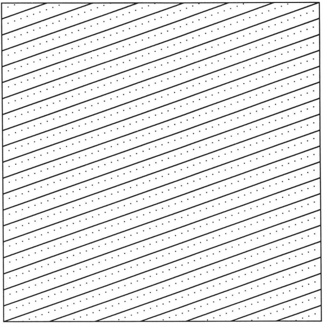
Arnold vermeldt ook een eenvoudig model waarbij wel sprake is van een goede menging, maar niet van echte chaos. Het betreft een torusafbeelding, of de afbeelding van een periodiek vierkant:

$$x \to x + a \quad \text{mod } 1, \qquad y \to y + b \quad \text{mod } 1$$

De punten van de banen liggen op de kronkellijn

$$y = (b/a)x + \text{constante}$$

Immers, telkens wanneer x een horizontale stap a maakt is y een verticale stap b verder gekomen. Is de verhouding b/a onmeetbaar, dan is de baan een kronkellijn die zich nooit sluit en de torus geheel overdekt. In gedachten denken we de baan geel gekleurd, zodat nu de hele torus geel is. Met een naburig beginpunt starten we een tweede blauwe baan. De tweede baan overdekt de torus eveneens maar blijft wel steeds op dezelfde afstand van zijn gele buurman. Het resultaat is een groene torus; de kleuren zijn goed gemengd, het dynamisch systeem voldoet aan de eis van goede menging. Toch is er geen chaos, omdat naburige banen hun onderlinge



figuur 16.3
Wel menging, geen chaos

verband blijven behouden. In figuur 16.3 is voor een periodiek vierkant die situatie nog eens grafisch weergegeven. De ene baan bestaat uit de getrokken schuine lijnen, eigenlijk een lange lijn die over de torus kronkelt. De tweede baan, gestippeld, blijft steeds op dezelfde afstand van de vorige.

Meten van chaos

In een dynamisch systeem kan de mate van chaos of orde op kwantitatieve wijze gezien worden als een groepje van een of meer karakteristieke getallen, *Lyapunov-getallen* of *Lyapunov-exponenten*, genoemd naar een Russisch wiskundige, tijdgenoot van Poincaré, die omstreeks 1900 actief was. In een eendimensionaal iteratief dynamisch systeem is er slechts een enkel Lyapunov-getal. Wordt het systeem beschreven door $x \rightarrow f(x)$, dan kunnen we de absolute waarde van de afgeleide van $f(x)$ opvatten als een lokale groeifactor. Is die factor kleiner dan 1 dan krimpt een klein lijnstukje bij een iteratiestap in, terwijl bij een groeifactor groter dan 1 het lijnstukje juist in lengte toeneemt.

Is $x_0, x_1, x_2, \ldots, x_n, \ldots$ een willekeurige baan die begint in x_0, dan kunnen we spreken over de gemiddelde groei van de baan als de gemiddelde

waarde (het meetkundig gemiddelde) van de groeifactoren langs de baan, de gemiddelde waarde van $|df/dx_k)|$ dus. Uiteraard hangt dat gemiddelde af van het gekozen beginpunt x_0. Maar bij veel systemen blijkt dat bijna alle banen toch hetzelfde groeigemiddelde opleveren, en dan kunnen we spreken over de gemiddelde groei van het dynamische systeem. De daaraan verbonden waarde is dan per definitie het Lyapunov-getal van het dynamische systeem. Een Lyapunov-getal kleiner dan 1 wijst op convergentie, op orde, op het bestaan van een aantrekkend dekpunt of een aantrekkende cyclus van dekpunten. Een Lyapunov-getal groter dan 1 wijst op divergentie, op het bestaan van een vreemde aantrekker, op chaos.

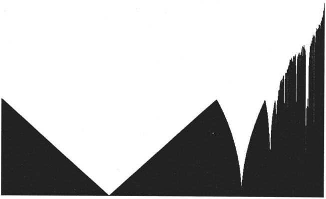
Kortom, het Lyapunov-getal is een maat van orde en van chaos, een maat voor de structuur in de chaos. In plaats van een Lyapunov-getal gebruikt men vaak de logaritme ervan als de zogenaamde Lyapunov-exponent. Een negatieve exponent wijst dan op orde, en een positieve op chaos.

Orde en chaos in het model van Verhulst

Om ons wat vertrouwd te maken met de nieuwe begrippen beschouwen we het bekende model van de geremde groei van Pierre Verhulst

$$x_{k+1} = ax_k(1 - x_k)$$

waarbij a tussen 0 en 4 ligt en x tot het interval (0,1) beperkt is. We voeren eens een experiment uit. De groeifactor in het punt x is gelijk aan $a|1 - 2x|$. Met behulp van het programma LYAPUN1 bepalen we voor een reeks waarden van a de gemiddelde groei uit een 'willekeurige' baan van duizend punten en zetten we de gevonden waarden grafisch uit als in figuur 16.4, waarbij a van 1 tot aan 4 loopt.



 figuur 16.4 De gemiddelde groei in het Verhulst-model

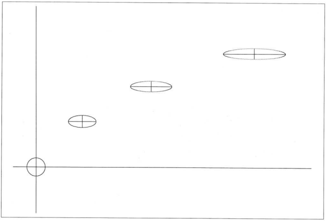
Het overeenkomstige programma LYAPUN2, dat maar heel weinig van het vorige programma verschilt, geeft de grafiek van de Lyapunov-getallen. Een kleine bijzonderheid is dat de waarde nul bij de gemiddelde groei, een superstabiel dynamisch gedrag, overeenkomt met een Lyapunov-getal van min oneindig. In de grafiek zien we hooguit een plaatselijke sprong in negatieve richting, afhankelijk van de numerieke precisie.

Wanneer het systeem ordelijk is kan de gemiddelde groei soms op eenvoudige wijze vooraf bepaald worden. Voor $1 < a < 3$ is er een aantrekkend dekpunt waar bijna alle banen naar toegaan. Dat dekpunt is $1\text{-}1/a$ en de groeifactor voor dat dekpunt is volgens de bovenstaande bepaling gelijk aan $|2\text{-}a|$. Inderdaad zien we in het computerexperiment dat voor de genoemde a-waarden de grafiek de vorm heeft van een hoofdletter V. Een kleine curiositeit is dat voor $a = 2$ de groei nul is, het geval van een superstabiel aantrekkend dekpunt. Wanneer $3 < a < 1+\sqrt{6}$, is er een stabiele 2-cyclus. Zijn de dekpunten p en q, dan is de gemiddelde groei gelijk aan het meetkundig gemiddelde van $|1 - 2p|$ en $|1 - 2q|$. Een betrekkelijk eenvoudige berekening leidt tot het resultaat $\sqrt{|a^2 - 2a - 4|}$, hetgeen grafisch vertaald kan worden in bogen van een cirkel en van een hyperbool. Daarna kunnen we alleen maar experimenteel verder komen. Tot aan de Feigenbaum-grens van 3.669 zijn er alleen maar stabiele cycli met een steeds hogere periode. Maar daarna wordt het pas echt de moeite waard. Theoretisch verwachten we tot aan $a = 4$ meestal chaotisch gedrag met kleine en heel kleine 'vensters' van het a-interval waarin het dynamisch systeem weer ordelijk is. Uiteraard botsen we al spoedig met de grenzen van nauwkeurigheid en computertijd, maar ook wanneer we niet al te precies zijn – en we sneller willen opschieten – zien we hoe onze verwachtingen door het experiment bevestigd worden. De gemiddelde groei springt almaar heen en weer tussen waarden groter dan 1 (chaos) en kleiner dan 1 (orde).

Wanneer voor een bepaalde waarde van a het systeem echt chaotisch is moet voor een zuivere bepaling van de gemiddelde groei een behoorlijk lange baan worden genomen. Een baan van duizend punten is dan eigenlijk onvoldoende, en men zou dan eerder naar een baan van honderdduizend punten of meer toe moeten, en dat kost computertijd. Hieruit blijkt wel dat voor computerexperimenten met betrekking tot de gemiddelde groei of de Lyapunov-getallen een snelle computer en veel tijd nodig is. Ook zal men telkens een afweging moeten maken betreffende de gewenste numerieke en/of grafische nauwkeurigheid.

Meer dimensies

De Lyapunov-getallen zijn vooral belangrijk voor dynamische systemen in meer dimensies, zowel continue als iteratieve. Het model van Lorenz is een voorbeeld van een continu systeem in drie dimensies. Ook voor zulke modellen gaat het om een soort gemiddelde groei, maar het is iets ingewikkelder. Concentreren we ons nog even op het model van Lorenz, een dynamisch systeem waarbij de banen eruit zien als een aaneenschakeling van lusvormige kronkels, dan kunnen we de groei langs een baan als volgt visualiseren. Beschouwen we in een baanpunt een heel klein bolletje en



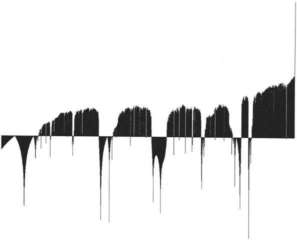
figuur 16.5
In een vlakke afbeelding zijn er twee Lyapunov-getallen

vervolgen we dat bolletje op zijn weg langs de baan, dan treedt in de afmetingen van dat bolletje weldra aan sterke vervorming op. In de ene richting wordt het volume-elementje uitgerekt terwijl in een andere richting verkorting kan optreden. Gaan we niet te ver dan heeft het bolletje bij benadering de vorm van een drieassige ellipsoïde aangenomen. Nu wordt de vorm van een dergelijke ellipsoïde gekenmerkt door de (halve) assen in drie hoofdrichtingen. We meten driemaal de verhouding van de lengte van een halve as tot die van de straal van het bolletje. Dat levert drie verhoudingsgetallen met de betekenis van plaatselijke groei. Er zijn dus in elk baanpunt drie lokale groeigetallen, even zoveel als het aantal dimensies van het systeem. Een middelingsproces langs een willekeurige baan, in dit voorbeeld de aantrekker van Lorenz, leidt daaruit drie Lyapunov-getallen af als meetgrootheden van gemiddelde groei. De Lyapunov-exponenten zijn daarvan weer de logaritmen. Ter verduidelijking lichten we in figuur 16.5 het principe toe van een tweetal Lyapunov-getallen voor een tweedimensionale afbeelding. We zien hoe een cirkeltje vervormd wordt tot een ellipsje als een samenspel van uitrekken en inkrimpen in verschillende richtingen.

Een goed voorbeeld van een tweedimensionale afbeelding is de bakkerstransformatie. De lokale groei kunnen we bepalen door te letten op de deformatie van een klein cirkeltje of vierkantje. Zonder rekenen zien we meteen dat het elementje in de x-richting met de factor 2 wordt uitgerekt en in de y-richting een inkrimping met dezelfde factor ondergaat. Een cirkeltje met straal r is dus een ellipsje geworden met halve assen $2r$ en $r/2$. De lokale groeigetallen zijn overal hetzelfde, zodat ook de gemiddelde groei met de twee Lyapunov-getallen 2 en 0.5 kan worden beschreven. De overeenkomstige Lyapunov-exponenten zijn log 2 en -log 2.

Bij de kat van Arnold is de berekening iets moeilijker omdat de vervorming van een cirkeltje een ellips in scheve stand tot gevolg heeft. Het berekenen van de hoofdassen vereist dan enige wiskundige techniek. Maar

figuur 16.6 De Lyapunov-getallen van het gemengde Verhulst-model voor $3.5 < a < 4$ en $b = 4$



ook in dat model is de lokale groei overal gelijk. Ten slotte vindt men dan de Lyapunov-getallen $(3 \pm \sqrt{5})/2$.

Zodra een der Lyapunov-getallen groter is dan 1 (of een Lyapunov-exponent positief is) is er chaotisch gedrag. Oppervlakkig gezien ziet elke vorm van chaos er eender uit, maar er zijn kennelijk zoveel soorten van chaos als er combinaties van Lyapunov-getallen zijn. Chaos kan dus gemeten worden.

Het model van Markus

We besluiten dit hoofdstuk met een moderne en interessante uitbreiding van het bekende model van Verhulst. Het model $x \to ax(1 - x)$ kan zoals bekend geïnterpreteerd worden als de relatieve groei van een insektenpopulatie met een constante groeifactor a. De Duitse onderzoeker Mario Markus heeft daaraan een even eenvoudige als interessante uitbreiding gegeven door de groeifactor a afhankelijk te stellen van het seizoen. Hij onderscheidt vruchtbare jaren waarin de groeifactor een hoge waarde heeft en slechte jaren waarin er een lage waarde is. De groeifactor wisselt dus volgens een bepaalde ritmiek tussen twee waarden, a en b genoemd. Markus kiest daarbij verschillende patronen zoals

$$ab, \quad aab, \quad aabab, \quad abaabbbaab$$

enzovoort.

Voor elke waardecombinatie van a en b kan het systeem zich ordelijk of chaotisch gedragen. We kunnen dan het Lyapunov-getal bepalen om beter inzicht te krijgen, maar omdat er nu opeens twee vrijheidsgraden zijn kunnen we de Lyapunov-getallen niet meer zo gemakkelijk als een functie

van (a,b) uitzetten. Een aantrekkelijke mogelijkheid is om het Lyapunov-getal van (a,b) volgens een bepaalde kleurencode als een gekleurde stip in het a,b-diagram weer te geven. Omdat er in de veelgebruikte VGA-modus plaats is voor zestien kleuren kunnen we ze ook allemaal (eventueel minus wit en zwart) gebruiken. Er kan dan, wel na veel rekentijd, een fraaie illustratie ontstaan als een voorbeeld van fantastische kunst, die kan dienen als cover voor een science-fictionverhaal.

Eenvoudiger is het echter eerst als een soort doorsnede van het a,b-diagram een grafiek te maken van de Lyapunov-getallen voor een vaste waarde van b en voor waarden van a uit een voorgeschreven interval. In het programma LYAPUN3 is dit gebeurd voor het patroon *ab* op de wijze van LAYPUN1 en LYAPUN2. Het resultaat zien we in figuur 16.6 voor $3.5 < a < 4$ en $b = 3$.

Hoewel de nauwkeurigheid niet erg groot is geeft het plaatje toch een aardige indruk van de grilligheid van het dynamische gedrag van het systeem. Wanneer de grafiek boven de nullijn is, is er chaos, maar de chaos wordt weer telkenmale afgewisseld door stabiel gedrag voor kleine venstertjes in het a-interval. In elk stabiel gebiedje verwachten we een a-waarde waarvoor het systeem superstabiel is. Er is dan nulgroei hetgeen een Lyapunov-getal van min oneindig (!) impliceert. In de grafiek merken we daar maar weinig van, daarvoor zijn de berekeningen te grof.

We eindigen met een korte bespreking van een a,b-diagram voor een model met het repeterend patroon *aabab* overeenkomstig het programma LYAPUN4. Dit programma is stellig een van de fraaiste, vergelijkbaar met programma's van details van de Mandelbrot set. Ook hier zijn kleuren essentieel en neemt de uitwerking veel rekentijd in beslag. Het programma kan gebruikt worden voor een willekeurig a-interval en een willekeurig b-interval. Men zou kunnen beginnen met voor beide parameters het interval (3,4) te kiezen. Met de getallen n_1 en n_2 kan een overeenkomstig deel van het beeldscherm worden afgezonderd, een rechthoek van $4n_1n_2$ pixels. Elke combinatie van a en b correspondeert dus met de positie van een pixel. Uit het Lyapunov-getal dat daarbij behoort kan een kleurwaarde worden afgeleid als een getal van 0 tot 15. In het programma is gekozen voor de afbeelding van de negatieve Lyapunov-getallen, de situaties van orde (figuur 16.7, bladzijde 88 in het kleurkatern).

De computerprogramma's

Alle 72 programma's van dit boek zijn hieronder afgedrukt. Om zoveel mogelijk programma's te kunnen plaatsen is de typografie enigszins aangepast en soms het programma wat ingekort. Zo ontbreken bijvoorbeeld de aanwijzingen die nodig zijn als het programma stopt. In dat geval dient men een willekeurige toets aan te slaan. En als men het programma wil beëindigen gebruikt men de toetsen X en Q.

Hoofdstuk 1	PYTKLEED	PRIEM	GAUSS	
Hoofdstuk 2	LISSA1	LISSA2	LISSA3	LISSAF
Hoofdstuk 3	HOOGTE1	HOOGTE2	LEVEL	PATROON1
	PATROON2	PATROON3	PATROON4	
Hoofdstuk 4	ZWEVING	HEK	MOIREL1	MOIREL2
	MOIREC1	MOIREC2	MOIRES1	MOIRES2
	MOIRES3	MOIRES4		
Hoofdstuk 5	POTVELD1	POTVELD2	POTVELD3	POTVELD4
Hoofdstuk 6	BOOMMC	BOOMBT	VAREN	
Hoofdstuk 7	MONDRIAN	ARTBLOK	KLEE	EI BLOKEI
	PAASEI			
Hoofdstuk 8	TURTLE	TURTLEK	TURTLEA	TURTLED
Hoofdstuk 9	PEANO	PEANOB		
Hoofdstuk 10	LIND1	LIND2		
Hoofdstuk 11	BOL	AARDE	EILAND	KAART
Hoofdstuk 12	PLANEET	SONDE	STARSHIP	
Hoofdstuk 13	KOMEET	PLANEET1	PLANEET2	
Hoofdstuk 14	HARM	POL1	POL2	POL3
Hoofdstuk 15	TORUS	KLOK	KLOKM	CIRKELH1
	CIRKELH2	CIRKELH4	CIRKELH5	
Hoofdstuk 16	BAKERB	BAKERP	LYAPUN1	LYAPUN2
	LYAPUN3	LYAPUN4		

Bestellen van de diskette 'simulatica'

De bovengenoemde programma's zijn samengebracht op een diskette. Deze diskette bevat nog een vijftigtal extra programma's, die aanvullingen en uitbreidingen bevatten van de in dit boek behandelde onderwerpen. De programma's op deze gebruikersdiskette zijn in het algemeen voorzien van meer commentaar en zijn ook meer op gebruikersgemak afgestemd. De diskette, type 3.5", kan bij de uitgever worden besteld door overmaking van f 37,50 op postbanknummer 58266 ten name van Aramith Uitgevers te Bloemendaal, onder vermelding van 'simulatica'.

Hoofdstuk 1

PYTKLEED

```
REM ***pythagoraskleed***
SCREEN 12 : CLS
XM=320 : YM=240 : MAX=200
LINE (XM-MAX,YM+MAX)-(XM+MAX,
                YM-MAX),,B
FOR M=0 TO MAX : N=0
DO WHILE N<=M AND INKEY$=""
X=M*M-N*N : Y=2*M*N
IF X<MAX AND Y<MAX THEN GOSUB graphics
N=N+1
LOOP
NEXT M : A$=INPUT$(1)
END
graphics:
PSET (XM+X,YM-Y) : PSET (XM+X,YM+Y)
PSET (XM-X,YM-Y) : PSET (XM-X,YM+Y)
PSET (XM+Y,YM-X) : PSET (XM+Y,YM+X)
PSET (XM-Y,YM-X) : PSET (XM-Y,YM+X)
RETURN
```

PRIEM

```
REM ***factorisatie getal <2 147 483 648***
SCREEN 12 : CLS
DEFLNG N
J=1 'initialisatie aantal factoren
PRINT"geef getal <2 147 483 648"
INPUT"getal = ",N : CLS
IF N=0 THEN END
T0=TIMER
REM ***factoren 2 ***
DO WHILE N/2=INT(N/2)
PRINT J,2 : J=J+1
N=INT(N/2)
LOOP
F=3
REM ***factoren 3 ***
DO WHILE N/3=INT(N/3)
PRINT J,3 : J=J+1
N=INT(N/3)
LOOP
REM ***volgende oneven factoren***
DO WHILE F<SQR(N)
F=F+2
DO WHILE N/F=INT(N/F)
PRINT J,F : J=J+1
N=INT(N/F)
LOOP
LOOP
T1=TIMER
IF N>1 THEN
PRINT J,N
PRINT : PRINT"het factoriseren duurde";
PRINT USING "###.##" ; T1-T0;
PRINT "seconden"
END IF
A$=INPUT$(1)
END
```

GAUSS

```
REM ***priemgetallenkleedje van Gauss***
SCREEN 12 : CLS
WINDOW (-80,-60)-(80,60)
N=40 'grootte van kleedje
LINE (-N-2,-N-2)-(N+2,N+2),,B
FOR N2=0 TO N : FOR N1=N2 TO N
S=SQR(N1*N1+N2*N2)
F2=0
DO UNTIL F2*F2>S
F1=0
DO UNTIL F1*F1+F2*F2>S
F=F1*F1+F2*F2
IF F<>0 AND F<>1 THEN
A=(N1*F1+N2*F2)/F : B=ABS(N1*F2-N2*F1)/F
IF A-INT(A)=0 AND B-INT(B)=0 THEN GOTO
                        nextpoint
END IF
F1=F1+1
LOOP
F2=F2+1
LOOP
REM ***graphics***
CIRCLE (N1,N2),.2 : CIRCLE (N2,N1),.2
CIRCLE (-N1,N2),.2 : CIRCLE (-N2,N1),.2
CIRCLE (-N1,-N2),.2 : CIRCLE (-N2,-N1),.2
CIRCLE (N1,-N2),.2 : CIRCLE (N2,-N1),.2
nextpoint:
NEXT N1 : NEXT N2 : A$=INPUT$(1)
END
```

Hoofdstuk 2

LISSA1

```
REM ***Lissajous krommen***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-2,-1.5)-(2,1.5)
DO
PRINT"stop het programma met de keuze s=0"
PRINT"geef de frequentieverhouding s"
INPUT"s = ",S : PRINT
IF S=0 THEN CLS : END
PRINT"geef de faseverschuiving als getal
                    tussen 0 en 1"
INPUT"f = ",F : CLS
A=1 'amplitude
F=2*PI*F
T=0 : H=.01 'initialisatie
DO WHILE INKEY$=""
X=A*COS(S*T)
Y=COS(T+F)
PSET (X,Y) : T=T+H
LOOP
LOOP
END
```

LISSA2

```
REM ***figuur van Lissajous***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-2.4,-1.8)-(2.4,1.8)
```

```
H=PI/100 : AMPL1=1.6 : AMPL2=1.2
DO
PRINT"kies de frequentie als een breuk van"
PRINT"twee gehele getallen m en n met m<n"
PRINT"een goede keuze is 8,13 of 34,55"
PRINT"stop het programma met m=0 of n=0"
INPUT"m = ",M : IF M=0 THEN END
INPUT"n = ",N : IF N=0 THEN END
PRINT"kies fase als getal tussen 0 en 1"
PRINT"een goede keuze is 0 of .25 of .75"
INPUT"fase = ",FASE
FREQ=M/N : FASE=FASE*2*PI
CLS
LINE (-1.1*AMPL1,-1.1*AMPL2)-
                    (1.1*AMPL1,1.1*AMPL2),,B
PSET (0,AMPL2*COS(FASE))
FOR K=0 TO 200*N : T=K*H
X=AMPL1*SIN(FREQ*T) :
Y=AMPL2*COS(T+FASE)
LINE -(X,Y)
NEXT K : A$=INPUT$(1) : CLS
LOOP
END
```

LISSA3

```
REM ***Lissajousachtige krommen***
SCREEN 12 : CLS
WINDOW (-2,-1.5)-(2,1.5)
PRINT"stop programma met een willekeurige toets"
A=1 : B=.5 : S=2.45 'parameters
T=0 : H=.01 'initialisatie
LINE (-A-B-.1,-1.1)-(A+B+.1,1.1),,B
DO WHILE INKEY$=""
X=A*COS(T)+B*COS(S*T)
Y=SIN(T)
IF T=0 THEN PSET (X,Y) ELSE LINE -(X,Y)
T=T+H
LOOP
LOCATE 1,1 : PRINT SPACE$(80)
END
```

LISSAF

```
REM ***Lissajousachtige kromme in de vorm
                        van een bloem***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-4,-3)-(4,3)
T=0 : H=.01
A1=1.5 : A2=.2 'te variëren parameters
S1=6 : S2=36 'parameters
T=0 : H=.01 'initialisatie
DO WHILE T<2*PI
R=1+A1*COS(S1*T)+A2*COS(S2*T)
PHI=T
X=R*COS(PHI) : Y=R*SIN(PHI)
IF T=0 THEN PSET (X,Y) ELSE LINE -(X,Y)
T=T+H
LOOP : A$=INPUT$(1)
END
```

Hoofdstuk 3

HOOGTE1

```
SCREEN 12 : CLS
XM=320 : YM=240 : N=160
M=10 'aantal niveau's
LINE (160,80)-(480,400),,B
FOR I =-N TO N : FOR J=-N TO N
IF INKEY$<>"" THEN END
X=I/N : Y=J/N
Z=X*X-Y*Y 'oppervlak
IF INT(M*Z) MOD 2=0 THEN PSET (XM+I,YM-J)
NEXT J : NEXT I
A$=INPUT$(1)
END
```

HOOGTE2

```
SCREEN 12 : CLS
XM=320 : YM=240 : N=160
M=10 'aantal niveau's
D=N+8
DIM COL(10)
DATA 0,1,9,2,10,3,11,4,12,5,13
FOR I=0 TO 10 : READ COL(I) : NEXT I
LINE (XM-D,YM-D)-(XM+D,YM+D),10,B
FOR I =-N TO N : FOR J=-N TO N
IF INKEY$<>"" THEN END
X=I/N : Y=J/N
Z=X*X-Y*Y 'oppervlak
L=1+INT(M*(1+Z)) MOD 10
PSET (XM+I,YM-J),COL(L)
NEXT J : NEXT I
A$=INPUT$(1)
END
```

LEVEL

```
REM ***level lines of z=(x,y) for -1<z<1***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-4.8,-3.6)-(4.8,3.6)
XS1=-PI : XS2=PI : YS1=-PI : YS2=PI
LINE (XS1,YS1)-(XS2,YS2),,B 'frame
DEF FNA(X,Y)=SIN(X*X+Y*Y-2*X+1)*
                    SIN(X*X+Y*Y+2*X+1)
IMAX=128 : JMAX=128
P=(XS2-XS1)/IMAX : Q=(YS2-YS1)/JMAX
KMAX=8 : DIM Z(KMAX) 'number of levels
FOR K=0 TO KMAX : Z(K)=-1+2*K/KMAX :
                    NEXT K
REM ***input multiswitch***
DIM TABLE(2,2,2)
DATA 1,7,4,6,11,9,3,10,2,5,12,8,13,14
DATA 13,8,12,5,2,10,3,9,11,6,4,7,1
FOR I1=0 TO 2
FOR I2=0 TO 2
FOR I3=0 TO 2
READ TABLE(I1,I2,I3)
NEXT I3 : NEXT I2 : NEXT I1
REM ***main program***
FOR I=0 TO IMAX-1 : FOR J=0 TO JMAX-1
X=XS1+P*I : Y=YS1+Q*J
```

```
IF INKEY$<>"" THEN END
D1=FNA(X,Y) : D2=FNA(X+P,Y)
D3=FNA(X,Y+Q) : D4=FNA(X+P,Y+Q)
FOR K=0 TO KMAX
IF Z(K)<D1 AND Z(K)<D2 AND Z(K)<D3 AND
                      Z(K)<D4 THEN GOTO repeat
IF Z(K)>D1 AND Z(K)>D2 AND Z(K)>D3 AND
                      Z(K)>D4 THEN GOTO repeat
XC=X+P/2 : YC=Y+Q/2
ZC=(D1+D2+D3+D4)/4-Z(K) : SC=1+SGN(ZC)
XA=X : YA=Y : XB=X+P : YB=Y : GOSUB
                      multiswitch
XA=X+P : YA=Y : XB=X+P : YB=Y+Q : GOSUB
                      multiswitch
XA=X+P : YA=Y+Q : XB=X : YB=Y+Q : GOSUB
                      multiswitch
XA=X+P : YA=Y+Q : XB=X : YB=Y : GOSUB
                      multiswitch
repeat:
NEXT K : NEXT J : NEXT I : A$=INPUT$(1)
END
setline:
LINE (U1,V1)-(U2,V2)
RETURN
multiswitch:
ZA=FNA(XA,YA)-Z(K) : ZB=FNA(XB,YB)-Z(K)
SA=1+SGN(ZA) : SB=1+SGN(ZB)
T=TABLE(SA,SB,SC)
SELECT CASE T
CASE 1
CASE 2
U1=XA-ZA*(XB-XA)/(ZB-ZA)
V1=YA-ZA*(YB-YA)/(ZB-ZA)
U2=XA-ZA*(XC-XA)/(ZC-ZA)
V2=YA-ZA*(YC-YA)/(ZC-ZA)
GOSUB setline
CASE 3
U1=XB-ZB*(XC-XB)/(ZC-ZB)
V1=YB-ZB*(YC-YB)/(ZC-ZB)
U2=XB-ZB*(XA-XB)/(ZA-ZB)
V2=YB-ZB*(YA-YB)/(ZA-ZB)
GOSUB setline
CASE 4
U1=XC-ZC*(XA-XC)/(ZA-ZC)
V1=YC-ZC*(YA-YC)/(ZA-ZC)
U2=XC-ZC*(XB-XC)/(ZB-ZC)
V2=YC-ZC*(YB-YC)/(ZB-ZC)
GOSUB setline
CASE 5
PSET (XA,YA)
CASE 6
PSET (XB,YB)
CASE 7
PSET (XC,YC)
CASE 8
U1=XA : V1=YA
U2=XC-ZC*(XB-XC)/(ZB-ZC)
V2=YC-ZC*(YB-YC)/(ZB-ZC)
GOSUB setline
CASE 9
U1=XB : V1=YB
U2=XA-ZA*(XC-XA)/(ZC-ZA)
V2=YA-ZA*(YC-YA)/(ZC-ZA)
GOSUB setline
CASE 10
U1=XC : V1=YC
U2=XB-ZB*(XA-XB)/(ZA-ZB)
V2=YB-ZB*(YA-YB)/(ZA-ZB)
GOSUB setline
CASE 11
LINE (XB,YB)-(XC,YC)
CASE 12
LINE (XC,YC)-(XA,YA)
CASE 13
LINE (XA,YA)-(XB,YB)
CASE 14
LINE (XB,YB)-(XC,YC)
LINE (XC,YC)-(XA,YA)
LINE (XA,YA)-(XB,YB)
END SELECT
RETURN
```

PATROON1

```
REM ***een vierkant patroon***
SCREEN 12 : CLS
XM=320 : YM=240 : N=50
INPUT"geef getal van 3 cijfers: ",C : CLS
FOR I=-N TO N : FOR J=-N TO N
IF INKEY$<>"" THEN END
X=I/N : Y=J/N
Z=(1-X*X)*(1-Y*Y) 'functiekeuze
IF INT(C*Z) MOD 2 = 0 THEN GOSUB graphics
NEXT J : NEXT I
LINE (XM-3*N-10,YM-3*N-10)-
                (XM+3*N+10,YM+3*N+10),,B
A$=INPUT$(1)
END
graphics:
PSET (XM+3*I,YM-3*J)
RETURN
```

PATROON2

```
REM ***rechthoekig kruissteekjespatroon***
SCREEN 12 : CLS
XM=320 : YM=240 : N1=40 : N2=30
M=5 : D=6 : H=3 : C=500
FOR I=0 TO N1 : FOR J=0 TO N2
IF INKEY$<>"" THEN END
X=I/N1 : Y=J/N2
Z=(1-X*X)*(1-Y*Y) 'functiekeuze
IF INT(C*Z) MOD M = 0 THEN GOSUB graphics
NEXT J : NEXT I
A$=INPUT$(1)
END
graphics:
X=XM+D*I : Y=YM-D*J : GOSUB kruis
X=XM+D*I : Y=YM+D*J : GOSUB kruis
X=XM-D*I : Y=YM-D*J : GOSUB kruis
X=XM-D*I : Y=YM+D*J : GOSUB kruis
RETURN
kruis:
PSET (X-H,Y-H) : LINE -(X+H,Y+H)
PSET (X-H,Y+H) : LINE -(X+H,Y-H)
RETURN
```

PATROON3

```
REM ***een patroon met bergen***
SCREEN 12 : CLS
XM=320 : YM=240 : N1=160 : N2=160 : C=30
DEF FNA(X,Y,A,B)=EXP(-A*X*X-B*Y*Y)
LINE (160,80)-(480,400),,B
FOR I=-N1 TO N1 : FOR J=-N2 TO N2
IF INKEY$<>"" THEN END
X=1.5*I/N1 : Y=1.5*J/N2
Z1=FNA(X-.6,Y-.3,5,5) : Z2=FNA(X+.6,Y-.3,5,5)
Z3=FNA(X,Y+.6,2.5,10)
Z=Z1+Z2+Z3+.1*(X*X+4*Y*Y) 'functiekeuze
IF INT(C*Z) MOD 2 = 0 THEN GOSUB graphics
NEXT J : NEXT I
A$=INPUT$(1) : END
graphics:
PSET (XM+I,YM-J)
RETURN
```

PATROON4

```
REM ***een rechthoekig patroon***
SCREEN 12 : CLS
XM=320 : YM=240 : N1=200 : N2=150 : C=1000
REM ***kleurentabel***
DIM COL(8)
DATA 0,1,9,2,10,4,12,6,14
FOR I=0 TO 8 : READ COL(I) : NEXT I
FOR I=0 TO N1 : FOR J=0 TO N2
IF INKEY$<>"" THEN END
X=I/N1 : Y=J/N2
Z=(1-X*X)*(1-Y*Y) 'functiekeuze
GOSUB graphics
NEXT J : NEXT I
X1=XM-N1-10 : Y1=YM-N2-10
X2=XM+N1+10 : Y2=YM+N2+10
LINE (X1,Y1)-(X2,Y2),14,B
A$=INPUT$(1)
END
graphics:
L=COL(1+INT(C*Z) MOD 8)
PSET (XM+I,YM-J),L : PSET (XM+I,YM+J),L
PSET (XM-I,YM-J),L : PSET (XM-I,YM+J),L
RETURN
```

Hoofdstuk 4

ZWEVING

```
REM ***zwevingen van golven***
SCREEN 12 : CLS
WINDOW (-.1,-6)-(1.1,6)
T=0 : PI=4*ATN(1) : S1=30 : S2=33
LINE (0,0)-(1,0)
DO WHILE T<1
Y1=SIN(2*S1*PI*T) : Y2=SIN(2*S2*PI*T)
PSET (T,Y1+Y2)
T=T+.0001
LOOP : A$=INPUT$(1)
END
```

HEK

```
REM ***zwevingen bij hekken***
REM ***naam:HEK***
SCREEN 12 : CLS
A1=19 : A2=22
H1=5 : H2=5
K=0 : B=100
DO WHILE K*A1+H1<620
FOR I=0 TO H1
LINE (20+I+K*A1,140)-(20+I+K*A1,150+B)
NEXT I : K=K+1
LOOP : K=0
DO WHILE K*A2+H2<620
FOR I=0 TO H2
LINE (20+I+K*A2,150)-(20+I+K*A2,160+B)
NEXT I : K=K+1
LOOP : A$=INPUT$(1)
END
```

MOIREL1

```
REM ***interferentie evenwijdige lijnen***
REM ***twee equidistante bundels***
SCREEN 12 : CLS
A=.3 : B=.32 : I=0
DO
X1=100 : Y1=80+I/A : X2=540 : Y2=Y1
LINE (X1,Y1)-(X2,Y2)
I=I+1
LOOP UNTIL Y1>400
J=0
DO
X1=100 : Y1=80+J/B : X2=540 : Y2=Y1
LINE (X1,Y1)-(X2,Y2)
J=J+1
LOOP UNTIL Y1>400
A$=INPUT$(1)
END
```

MOIREL2

```
REM ***interferentie evenwijdige lijnen***
REM ***tweede bundel iets gedraaid***
SCREEN 12 : CLS
I=0 : ALFA=.02 : C=COS(ALFA) : S=SIN(ALFA)
DO
X1=100 : Y1=80+4*I : X2=540 : Y2=Y1
LINE (X1,Y1)-(X2,Y2)
U1=C*X1-S*Y1 : V1=S*X1+C*Y1
U2=C*X2-S*Y2 : V2=S*X2+C*Y2
LINE (U1,V1)-(U2,V2)
I=I+1
LOOP UNTIL Y1>400
A$=INPUT$(1)
END
```

MOIREC1

```
REM ***interferentie van cirkelbundels***
REM ***naam:MOIREC1***
SCREEN 12 : CLS
YM=240 : PI=4*ATN(1) : H=.3
```

```
FOR M=1 TO 2
IF M=1 THEN XM=300 ELSE XM=340
FOR I=0 TO 40
IF INKEY$<>"" THEN END
R=1+4*I
S=0 : X=XM+R : Y=YM : PSET (X,Y)
DO
S=S+H/R
LINE -(XM+R*COS(S),YM+R*SIN(S))
LOOP UNTIL S>2*PI
NEXT I
IF M=1 THEN
PRINT"druk op een toets"
A$=INPUT$(1) : LOCATE 1,1
PRINT SPACE$(18)
END IF
NEXT M : LOCATE 1,1
PRINT"einde, druk op een toets"
END
```

MOIREC2

```
REM ***interferentie van cirkelbundels***
SCREEN 12 : CLS
YM=240 : PI=4*ATN(1) : H=.3
FOR M=1 TO 2
IF M=1 THEN XM=300 ELSE XM=320
FOR I=5 TO 60
IF INKEY$<>"" THEN END
R=24*SQR(I)
S=0 : X=XM+R : Y=YM : PSET (X,Y)
DO
S=S+H/R
LINE -(XM+R*COS(S),YM+R*SIN(S))
LOOP UNTIL S>2*PI
NEXT I
IF M=1 THEN
PRINT"druk op een toets"
A$=INPUT$(1) : LOCATE 1,1
PRINT SPACE$(18)
END IF
NEXT M : LOCATE 1,1
PRINT"einde, druk op een toets"
END
```

MOIRES1

```
REM ***interferentie evenwijdige lijnen***
REM ***met pixelrooster van beeldscherm***
SCREEN 12 : CLS
A=7 'verticale verplaatsing
FOR I=1 TO 200
X1=100 : Y1=50+2*I : X2=540 : Y2=A+Y1
LINE (X1,Y1)-(X2,Y2)
NEXT I : A$=INPUT$(1)
END
```

MOIRES2

```
REM ***interferentie van stralenbundel***
REM ***met pixelrooster van beeldscherm***
SCREEN 12 : CLS
X0=320 : Y0=0 'centrum bundel
FOR I=1 TO 160
X=4*I : Y=480
LINE (X0,Y0)-(X,Y)
NEXT I : A$=INPUT$(1)
END
```

MOIRES3

```
REM ***interferentie van cirkels met***
REM ***pixelrooster van beeldscherm***
REM ***stralen in lineaire progressie***
SCREEN 12 : CLS
XM=320 : YM=240 : PI=4*ATN(1) : H=.2
FOR I=10 TO 90
IF INKEY$<>"" THEN END
R=1+2*I
S=0 : X=XM+R : Y=YM : PSET (X,Y)
DO
S=S+H/R
LINE -(XM+R*COS(S),YM+R*SIN(S))
LOOP UNTIL S>2*PI
NEXT I : A$=INPUT$(1)
END
```

MOIRES4

```
REM ***interferentie van cirkelbogen met***
REM ***pixelrooster van beeldscherm***
SCREEN 12 : CLS
WINDOW (0,0)-(4,3)
MAX=400 : H=.0015
XM=-4 : YM=-1 'middelpunt cirkelbundel
R1=SQR(XM*XM+YM*YM)
R2=SQR((XM-4)*(XM-4)+(YM-3)*(YM-3))
FOR I=0 TO MAX
IF INKEY$<>"" THEN END
R=R1+(R2-R1)*I/MAX
F=-ATN(YM/SQR(R*R-YM*YM))
DO
X=XM+R*COS(F) : Y=YM+R*SIN(F)
PSET (X,Y)
F=F+H
LOOP UNTIL Y>4 OR X<-.1 OR F>1.6
NEXT I : A$=INPUT$(1)
END
```

Hoofdstuk 5

POTVELD1

```
REM ***potentiaallijnen en stroomlijnen***
REM ***van een puntbron***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
PI=4*ATN(1) : H=.2 : K=-4
DO
R=EXP(-K*H)
CIRCLE (0,0),R
K=K+1
LOOP UNTIL R<.2
R=2.5
FOR K=0 TO 7
```

```
PHI=K*PI/4
LINE (0,0)-(R*COS(PHI),R*SIN(PHI))
NEXT K : A$=INPUT$(1)
END
```

POTVELD2

```
REM ***potentiaallijnen en stroomlijnen***
REM ***van put en bron***
SCREEN 12 : CLS
WINDOW (-6,-4.5)-(6,4.5)
PSET (-1,0) : PSET (1,0)
PI=4*ATN(1) : H=.5
FOR K=-8 TO 7
C=EXP(K*H+H/2) : A=(1+C)/(1-C)
R=2*SQR(C)/ABS(1-C) : CIRCLE (A,0),R
NEXT K
FOR L=-4 TO 4
F=L*PI/10
IF L<>0 THEN B=1/TAN(F) : CIRCLE
(0,B),SQR(1+B*B)
NEXT L : A$=INPUT$(1)
END
```

POTVELD3

```
REM ***potentiaallijnen van twee***
REM ***gelijke puntbronnen***
REM ***naam:POTVELD3***
SCREEN 12 : CLS
WINDOW (-2,-1.5)-(2,1.5)
PI=4*ATN(1) : H=.2 : E=.00001
FOR K=0 TO 8
C=EXP(H*K)+E
FOR I=0 TO 800
T=2*I*PI/800
R=SQR(COS(2*T)+SQR(C-SIN(2*T)^2))
PSET (R*COS(T),R*SIN(T))
NEXT I
NEXT K
FOR L=1 TO 8 : C=EXP(-H*L)
FOR I=0 TO 200 : T=I*PI/200
X=SQR(1+C*COS(T))
Y=SQR(-2-
C*COS(T)+SQR(4+4*C*COS(T)+C*C)+E)
PSET (X,Y) : PSET (X,-Y)
PSET (-X,Y) : PSET (-X,-Y)
NEXT I
NEXT L : A$=INPUT$(1)
END
```

POTVELD4

```
REM ***potentiaallijnen van een***
REM *** regelmatige m-hoek***
SCREEN 12 : CLS
WINDOW (-2.4,-1.8)-(2.4,1.8)
PI=4*ATN(1) : M=5 : H=.2
FOR K=-4 TO 8
C=EXP(.001+H*K)
IF K>0 THEN GOSUB pot1 ELSE GOSUB pot2
NEXT K : A$=INPUT$(1)
END
pot1:
FOR I=0 TO 1000
T=I*M*PI/500
S=1+2*C*COS(T)+C*C : R=S^(1/(2*M))
F=(T-ATN(SIN(T)/(C+COS(T))))/M
PSET (R*COS(F),R*SIN(F))
NEXT I
RETURN
pot2:
FOR I=0 TO M-1
FOR J=0 TO 200
T=J*PI/100
S=1+2*C*COS(T)+C*C : R=S^(1/(2*M))
F=(ATN(C*SIN(T)/(1+C*COS(T)))+2*I*PI)/M
PSET (R*COS(F),R*SIN(F))
NEXT J
NEXT I
RETURN
```

Hoofdstuk 6

BOOMMC

```
REM ***zelfgelijkvormige fractal***
SCREEN 12 : CLS : RANDOMIZE 11
WINDOW (-2.4,-1.2)-(2.4,2.4)
NMAX=60000
REM ***coëfficienten***
A=.5 : B=.5 : C=.5 : D=-.5
DET1=A*A+B*B : DET2=C*C+D*D
Q=DET1/(DET1+DET2)
X=1 : Y=0 : N=0 'start
DO WHILE N<NMAX AND INKEY$=""
R=RND
IF R<Q THEN
X1=A*X-B*Y-1+A : Y1=B*X+A*Y+B
ELSE
X1=C*X-D*Y+1-C : Y1=D*X+C*Y-D
END IF
X=X1 : Y=Y1
PSET (X,Y),10
N=N+1
LOOP : A$=INPUT$(1)
END
```

BOOMBT

```
REM ***kale Pythagoras boom***
REM ***orde bepaald door kleinste takje***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-2.8,-1.8)-(2.8,2.4)
EPS=.01 'drempelwaarde
P=32 : DIM XA(P),YA(P),XB(P),YB(P),S(P)
REM ***initialisatie***
ALFA=PI/3 : BETA=PI/4 'hoek links en rechts
CA=.7 : CB=.7 'contractie links en rechts
A1=COS(ALFA) : B1=SIN(ALFA)
A2=COS(BETA) : B2=SIN(BETA)
XA=0 : YA=-1 : XB=0 : YB=0 : G=0 : M=0
LINE (0,-1)-(0,0)
DO WHILE M>=0
DIFF=ABS(XB-XA)+ABS(YB-YA)
IF G<P AND DIFF>EPS THEN
```

```
M=M+1 : G=G+1 : S(M)=G 'op stapel
GOSUB graphics
XA=XAN : YA=YAN : XB=XBN : YB=YBN
ELSE
XA=XA(M) : YA=YA(M) : XB=XB(M) : YB=YB(M)
G=S(M) : M=M-1 'van stapel
END IF
LOOP : A$=INPUT$(1)
END
graphics:
U=XB-XA : V=YB-YA
XAN=XB : YAN=YB
XBN=XB+CA*(A1*U-B1*V)
YBN=YB+CA*(B1*U+A1*V)
XA(M)=XB : YA(M)=YB
XB(M)=XB+CB*(A2*U+B2*V)
YB(M)=YB+CB*(-B2*U+A2*V)
LINE (XA,YA)-(XB,YB)
RETURN
```

VAREN

```
REM ***varenblad volgens Barnsley***
SCREEN 12 : CLS : RANDOMIZE 11
WINDOW (-1,-4.5)-(11,4.5)
NMAX=40000
REM ***coëfficienten***
D1=.16
A2=.85 : B2=.04 : C2=-.04 : D2=.85 : F2=1.6
A3=.2 : B3=-.26 : C3=.23 : D3=.22 : F3=1.6
A4=-.15 : B4=.28 : C4=.26 : D4=.24 : F4=.44
REM ***waarschijnlijkheidsverdeling***
Q1=.01 : Q2=.86 : Q3=.93
X=0 : Y=0 : N=0 'start
DO WHILE N<NMAX AND INKEY$=""
R=RND
SELECT CASE R
CASE <Q1
X=0 : Y=D1*Y
CASE Q1 TO Q2
Z=X : X=A2*X+B2*Y : Y=C2*Z+D2*Y+F2
CASE Q2 TO Q3
Z=X : X=A3*X+B3*Y : Y=C3*Z+D3*Y+F3
CASE >Q3
Z=X : X=A4*X+B4*Y : Y=C4*Z+D4*Y+F4
END SELECT
PSET (Y,X),10
N=N+1
LOOP : A$=INPUT$(1)
END
```

Hoofdstuk 7

MONDRIAN

```
REM ***moderne kunst***
SCREEN 12 : CLS
RANDOMIZE 11
WINDOW (-.7,-.4)-(1.7,1.4)
N=100 : H=8 'aantal lijntjes en lengte
LINE (-.15,-.15)-(1.15,1.15),,B
FOR K=1 TO N
V=H*(1-SQR(RND))/50
X=RND : Y=RND
U1=X-V : V1=Y : U2=X+V : V2=Y
U3=X : V3=Y-V : U4=X : V4=Y+V
IF RND<.5 THEN LINE (U1,V1)-(U2,V2)
ELSE
LINE (U3,V3)-(U4,V4)
END IF
NEXT K : A$=INPUT$(1)
END
```

ARTBLOK

```
REM ***moderne kunst***
REM ***voor Q(uick) Basic kieze men EGA***
CLS : PRINT"kies 9 voor EGA of 12 voor VGA"
INPUT"screengetal = ",SCR
SCREEN SCR : CLS
WINDOW (-.5, -.2)-(1.5, 1.3)
RANDOMIZE TIMER
M=16 'm*m blokken
Q=.9 'vullingsgraad
A=1.3 : P=A/(2*M) 'grootte blokken
NUMCOL=13 : DIM COL%(16) 'aantal kleuren
FOR I=0 TO NUMCOL : COL%(I)=I : NEXT I
NMAX=INT(M*M*ABS(LOG(1-Q)))
FOR K=0 TO NMAX
I=1+INT(M*RND) : J=1+INT(M*RND)
L=1+INT(100*RND) MOD NUMCOL
X1=I/M-P : Y1=J/M-P : X2=I/M+P : Y2=J/M+P
LINE (X1,Y1)-(X2,Y2),COL%(L),BF
NEXT K
LINE (1/M-2*P,1/M-2*P)-(1+2*P,1+2*P),14, B
LOCATE 1,1 : PRINT "toets X for einde,
                andere toetsen voor kleur"
repeat:
A$=INPUT$(1)
IF A$="X" OR A$="x" THEN END
GOSUB othercolours
GOTO repeat
END
othercolours:
FOR K=1 TO NUMCOL
COL%(K)=1+INT(SQR(3)*COL%(K)) MOD 15
PALETTE K,COL%(K)
NEXT K
RETURN
```

KLEE

```
REM ***moderne kunst***
SCREEN 12 : CLS
RANDOMIZE 111 : PI=4*ATN(1)
WINDOW (-.7,-.4)-(1.7,1.4)
N=40 : H=.05 'aantal figuurtjes
LINE (-2*H,-2*H)-(1+2*H,1+2*H),14,B
FOR K=1 TO N
V=H*(.5+RND) 'random grootte
X=RND : Y=RND
IF RND<.5 THEN GOSUB lijn ELSE
                GOSUB driehoek
NEXT K : A$=INPUT$(1)
END
lijn:
ALFA=RND*PI : A=COS(ALFA)
```

```
B=SIN(ALFA) : C=9+INT(4*RND)
U1=X-A*V : V1=Y-B*V : U2=X+A*V : V2=Y+B*V
LINE (U1,V1)-(U2,V2),C
RETURN
driehoek:
ALFA=2*RND*PI : C=1+INT(4*RND)
A1=COS(ALFA) : A2=COS(ALFA+2*PI/3)
A3=COS(ALFA-2*PI/3)
B1=SIN(ALFA) : B2=SIN(ALFA+2*PI/3)
B3=SIN(ALFA-2*PI/3)
U1=X+A1*V : U2=X+A2*V : U3=X+A3*V
V1=Y+B1*V : V2=Y+B2*V : V3=Y+B3*V
LINE (U1,V1)-(U2,V2),C
LINE -(U3,V3),C : LINE -(U1,V1),C
U1=X+.8*A1*V : U2=X+.8*A2*V : U3=X+.8*A3*V
V1=Y+.8*B1*V : V2=Y+.8*B2*V : V3=Y+.8*B3*V
PAINT (U1,V1),C : PAINT (U2,V2),C
PAINT (U3,V3),C
RETURN
```

EI

```
REM ***de vorm van een ei***
SCREEN 12 : CLS
WINDOW (-2,-1.5)-(2,1.5)
A=.7 : B=.2  'ei parameters
FOR K=0 TO 10 : C=1-K/12
PSET (-C,0)
FOR I=-200 TO 200 : X=I/200
Z=(1-X*X)*(1+B*X) : Y=A*SQR(Z)
LINE -(C*X,C*Y)
NEXT I
PSET (-C,0)
FOR I=-200 TO 200 : X=I/200
Z=(1-X*X)*(1+B*X) : Y=A*SQR(Z)
LINE -(C*X,-C*Y)
NEXT I
NEXT K : A$=INPUT$(1)
END
```

BLOKEI

```
REM ***gekleurd paasei***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-1.2,-.9)-(1.2,.9)
NUMCOL=12  'aantal kleuren
P=100  'kleurparameter
A=.8  'verticale verkorting
DEF FNB(X,Y)=A*SQR(1-X*X)*SIN(PI*Y/2)
NHOR=4 : NVER=3 : NH=2*NHOR+1 :
                    NV=2*NVER+1
FOR I=-NHOR TO NHOR :
                 FOR J=-NVER TO NVER
X0=2*I/NH : Y0=2*J/NV
C=1+INT(P*RND) MOD NUMCOL
GOSUB trafo
PAINT (X0,FNB(X0,Y0)),C
NEXT J : NEXT I
END
trafo:
FOR K=0 TO 200
X=X0-1/NH : Y=Y0-1/NV+K/(NV*100)
Z=FNB(X,Y)
IF K=0 THEN PSET (X,Z),C ELSE LINE -(X,Z),C
NEXT K
FOR K=0 TO 200
Y=Y0+1/NV : X=X0-1/NH+K/(NH*100)
LINE -(X,FNB(X,Y)),C
NEXT K
FOR K=0 TO 200
X=X0+1/NH : Y=Y0+1/NV-K/(NV*100)
LINE -(X,FNB(X,Y)),C
NEXT K
FOR K=0 TO 200
Y=Y0-1/NV : X=X0+1/NH-K/(NH*100)
LINE -(X,FNB(X,Y)),C
NEXT K
RETURN
```

PAASEI

```
REM ***een eivormige aantrekker van een***
REM ***dynamisch systeem***
REM ***voor Q(uick) Basic kieze men EGA"
CLS : PRINT"kies 9 voor EGA of 12 voor VGA"
INPUT"screengetal = ", SCR
SCREEN SCR : CLS
WINDOW (-.4,-.3)-(.4,.3)
NUMCOL=8 : DIM COL%(NUMCOL)
FOR I=0 TO NUMCOL : COL%(I)=I : NEXT I
A=.996 : B=.102  'rotatieparameters
P=.05 : Q=.2  'systeemparameters
R=.0025  'grootte van cirkeltje
X=.04 : Y=0  'beginpunt van baan
FOR N=0 TO 3000
IF INKEY$<>"" THEN END
K=1+N MOD NUMCOL
CIRCLE (X,Y),R,K : PAINT (X,Y),K
S=1+P*X*X-Q*Y
X1=(A*X-B*Y/.7)/S : Y1=B*X*.7+A*Y
X=X1 : Y=Y1
NEXT N
repeat:
LOCATE 1,1
PRINT"toets X voor einde, andere toetsen
                          voor kleur"
A$=INPUT$(1)
IF A$="X" OR A$="x" THEN END
GOSUB othercolours
GOTO repeat
END
othercolours:
FOR K=1 TO NUMCOL
COL%(K)=1+INT(SQR(5)*COL%(K)) MOD 63
PALETTE K,COL%(K)
NEXT K
RETURN
```

Hoofdstuk 8

TURTLE

```
REM ***turtlegraphics, stap voor stap***
SCREEN 12 : CLS
XM=320 : YM=240 : H=10
PRINT"F voorwaarts, + kwartslag linksom,
              - kwartslag rechtsom, x einde"
```

```
X=XM : Y=YM : P=1 : Q=0 : PSET (X,Y)
DO
A$=INPUT$(1)
SELECT CASE A$
CASE "F","f"
X=X+H*P : Y=Y+H*Q
LINE -(X,Y)
CASE "+"
P1=Q : Q1=-P
P=P1 : Q=Q1
CASE "-"
P1=-Q : Q1=P
P=P1 : Q=Q1
END SELECT
LOOP UNTIL A$="X" OR A$="x"
END
```

TURTLEK

```
REM ***de lijn van Von Koch***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
NMAX=4 : PI=4*ATN(1)
REM ***beginpositie en staplengte***
X=-3 : Y=-.5 : H=.075
REM ***axioma en productieregel***
AXIOM$="F"
PROD$="F+F—F+F"
REM ***vorming van woord***
WEG$=AXIOM$
FOR N=1 TO NMAX
W$=""
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
IF S$="F" THEN Q$=PROD$ ELSE Q$=S$
W$=W$+Q$
NEXT I
WEG$=W$
NEXT N
REM ***graphics***
A=COS(PI/3) : B=SIN(PI/3)
P=1 : Q=0
PSET (X,Y)
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
SELECT CASE S$
CASE "+"
P1=A*P-B*Q : Q1=B*P+A*QP=P1 : Q=Q1
CASE "-"
P1=A*P+B*Q : Q1=-B*P+A*Q
P=P1 : Q=Q1
CASE "F"
X=X+H*P : Y=Y+H*Q
LINE -(X,Y)
END SELECT
NEXT I : A$=INPUT$(1)
END
```

TURTLEA

```
REM ***Mandelbrot's archipel***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
NMAX=2 : PI=4*ATN(1)
REM ***beginpositie en staplengte***
X=-1 : Y=1 : PHI=0 : H=.06
REM ***axioma en productieregel***
AXIOM$="F-F-F-F"
PROD1$="F-f+FF-F-FF-Ff-FF+f-
                    FF+F+FF+Ff+FFF"
PROD2$="ffffff"
REM ***vorming van woord***
WEG$=AXIOM$
FOR N=1 TO NMAX
W$=""
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
IF S$="F" THEN Q$=PROD1$
IF S$="f" THEN Q$=PROD2$
IF S$="+" OR S$="-" THEN Q$=S$
W$=W$+Q$
NEXT I
WEG$=W$
NEXT N
REM ***graphics***
PSET (X,Y)
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
SELECT CASE S$
CASE "+"
PHI=PHI+PI/2
CASE "-"
PHI=PHI-PI/2
CASE "F"
X=X+H*COS(PHI) : Y=Y+H*SIN(PHI)
LINE -(X,Y)
CASE "f"
X=X+H*COS(PHI) : Y=Y+H*SIN(PHI)
PSET (X,Y),0
END SELECT
NEXT I : A$=INPUT$(1)
END
```

TURTLED

```
REM ***de drakenlijn***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
NMAX=12 : PI=4*ATN(1)
REM ***beginpositie en staplengte**
X=1.5 : Y=.5 : H=.06
REM ***axioma en productieregels***
AXIOM$="F"
PROD1$="F-G" : PROD2$="F+G"
REM ***vorming van woord***
LOCATE 1,1 : PRINT"please wait ..."
WEG$=AXIOM$
FOR N=1 TO NMAX
W$=""
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
IF S$="F" THEN Q$=PROD1$
IF S$="G" THEN Q$=PROD2$
IF S$="+" OR S$="-" THEN Q$=S$
W$=W$+Q$
NEXT I
WEG$=W$
NEXT N
REM ***graphics***
```

```
P=1 : Q=0
PSET (X,Y)
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
SELECT CASE S$
CASE "+"
P1=-Q : Q1=P
P=P1 : Q=Q1
CASE "-"
P1=Q : Q1=-P
P=P1 : Q=Q1
CASE "F","G"
X=X+H*P : Y=Y+H*Q
LINE -(X,Y)
END SELECT
NEXT I : LOCATE 1,1
PRINT SPACE$(80) : A$=INPUT$(1)
END
```

Hoofdstuk 9

PEANO

```
REM ***Peano lijn volgens Hilbert***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
NMAX=3 : PI=4*ATN(1)
REM ***beginpositie en staplengte***
X=-2 : Y=2 : PHI=0 : H=.6
REM ***axioma en productieregels***
AXIOM$="X"
PROD1$="-YF+XFX+FY-"
PROD2$="+XF-YFY-FX+"
REM ***hoofdprogramma***
WEG$=AXIOM$
FOR N=1 TO NMAX
W$=""
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
SELECT CASE S$
CASE "X"
Q$=PROD1$
CASE "Y"
Q$=PROD2$
CASE ELSE
Q$=S$
END SELECT
W$=W$+Q$
NEXT I
WEG$=W$
NEXT N
REM ***graphics***
PSET (X,Y)
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
SELECT CASE S$
CASE "+"
PHI=PHI+PI/2
CASE "-"
PHI=PHI-PI/2
CASE "F"
X=X+H*COS(PHI) : Y=Y+H*SIN(PHI)
LINE -(X,Y)
END SELECT
NEXT I : A$=INPUT$(1)
END
```

PEANOB

```
REM ***Peano lijn volgens Hilbert***
REM ***versie met backtracking***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
DEFINT L,N,S
DEFDBL H,P,X,Y
NMAX=6 : PI=4*ATN(1)
REM ***beginpositie en staplengte***
X=-2.2 : Y=2.5 : PHI=0 : H=.07
REM ***axioma en productieregels***
AXIOM$="X"
PROD1$="-YF+XFX+FY-"
PROD2$="+XF-YFY-FX+"
L1=LEN(PROD1$) : L2=LEN(PROD2$)
IF L1>L2 THEN LP=L1 ELSE LP=L2
DIM P$(LP*NMAX),N(LP*NMAX)
REM ***hoofdprogramma***
S=1 : N=0 : P$(1)="X" : N(1)=0
DO WHILE S>0
DO WHILE N<NMAX
S$=P$(S) : N=N(S)+1 : S=S-1
SELECT CASE S$
CASE "X"
FOR I=0 TO LP-1
M$=MID$(PROD1$,LP-I,1)
S=S+1
P$(S)=M$ : N(S)=N
NEXT I
CASE "Y"
FOR I=0 TO LP-1
M$=MID$(PROD2$,LP-I,1)
S=S+1
P$(S)=M$ : N(S)=N
NEXT I
CASE "F","+","-"
S=S+1 : P$(S)=S$ : N(S)=N
END SELECT
LOOP
S$=P$(S) : GOSUB graphics
S=S-1 : N=N(S)
LOOP : A$=INPUT$(1)
END
graphics:
PSET (X,Y)
SELECT CASE S$
CASE "+"
PHI=PHI+PI/2
CASE "-"
PHI=PHI-PI/2
CASE "F"
X=X+H*COS(PHI) : Y=Y+H*SIN(PHI)
LINE -(X,Y)
END SELECT
RETURN
```

Hoofdstuk 10

LIND1

```
REM ***plantengroei volgens Lindenmayer***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
PI=4*ATN(1) : NMAX=5
DIM U(16),V(16),W(16)
REM ***beginpositie, hoek en staplengte***
X=0 : Y=-2.5 : ALFA=PI/7 : H=.02
REM ***axioma en productieregel***
AXIOM$="F"
PROD$="F[+F]F[-F]F"
REM ***vorming van woord***
WEG$=AXIOM$
FOR N=1 TO NMAX
W$=""
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
IF S$="F" THEN Q$=PROD$ ELSE Q$=S$
W$=W$+Q$
NEXT I
WEG$=W$
NEXT N
REM ***graphics***
PHI=PI/2 : M=1
PSET (X,Y)
FOR I=1 TO LEN(WEG$)
S$=MID$(WEG$,I,1)
SELECT CASE S$
CASE "+"
PHI=PHI+ALFA
CASE "-"
PHI=PHI-ALFA
CASE "F"
X=X+H*COS(PHI) : Y=Y+H*SIN(PHI)
LINE -(X,Y)
CASE "["
M=M+1 : U(M)=X : V(M)=Y : W(M)=PHI
CASE "]"
X=U(M) : Y=V(M) : PHI=W(M) : M=M-1
PSET (X,Y)
END SELECT
NEXT I
END
```

LIND2

```
REM ***plantengroei volgens Lindenmayer***
REM ***versie met backtracking***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
NMAX=6 : X=0 : Y=-2.5 : PI=4*ATN(1)
PHI=PI/2 : DELTA=PI/8 : H=.032
REM ***axioma en productieregel***
AXIOM$="X"
PROD1$="F-[[X]+X]+F[+FX]-X"
PROD2$="FF"
LP1=LEN(PROD1$) : LP2=LEN(PROD2$)
LP=LP1+LP2
DIM P$(LP*NMAX),N(LP*NMAX)
DIM U(16),V(16),W(16)
REM ***hoofdprogramma
S=1 : N=0 : P$(1)="X" : N(1)=0 : M=0
DO WHILE S>0
DO WHILE N<NMAX
N=N(S)+1
SELECT CASE P$(S)
CASE "X"
S=S-1
FOR I=LP1 TO 1 STEP -1
S=S+1 : N(S)=N
P$(S)=MID$(PROD1$,I,1)
NEXT I
CASE "F"
S=S-1
FOR I=LP2 TO 1 STEP -1
S=S+1 : N(S)=N
P$(S)=MID$(PROD2$,I,1)
NEXT I
CASE "+","-","[","]"
N(S)=N
END SELECT
LOOP
GOSUB graphics : S=S-1 : N=N(S)
LOOP : A$=INPUT$(1)
END
graphics:
PSET (X,Y)
SELECT CASE P$(S)
CASE "+"
PHI=PHI+DELTA
CASE "-"
PHI=PHI-DELTA
CASE "F"
X=X+H*COS(PHI) : Y=Y+H*SIN(PHI)
LINE -(X,Y)
CASE "["
M=M+1 : U(M)=X : V(M)=Y : W(M)=PHI
U=X : V=Y : W=PHI
CASE "]"
X=U(M) : Y=V(M) : PHI=W(M) : M=M-1
PSET (X,Y)
END SELECT
RETURN
```

Hoofdstuk 11

BOL

```
REM ***bol, breedtecirkels en meridianen***
SCREEN 12 : CLS
WINDOW (-2,-1.5)-(2,1.5)
C=1/SQR(2) : PI=4*ATN(1) : PHI=25
SF=SIN(PHI*PI/180) : CF=COS(PHI*PI/180)
CIRCLE (0,0),1
REM ***breedtecirkels***
FOR I=-5 TO 5 : B=PI*I/12
SB=SIN(B) : CB=COS(B)
FOR J=0 TO 360 STEP 2 : T=J*PI/180
X=COS(T)*CB : Y=SIN(T)*CB : Z=SB
U=C*(Y-X)
V=-C*SF*(X+Y)+CF*Z W=C*CF*(X+Y)+SF*Z
IF W>0 THEN PSET (U,V)
NEXT J
NEXT I
CIRCLE (0,0),1
```

```
REM ***meridianen***
FOR I=0 TO 5 : A=PI*I/6
SA=SIN(A) : CA=COS(A)
FOR J=0 TO 360 STEP 2 : T=J*PI/180
X=CA*COS(T) : Y=SA*COS(T) : Z=SIN(T)
U=C*(Y-X) : V=-C*SF*(X+Y)+CF*Z
W=C*CF*(X+Y)+SF*Z
IF W>0 THEN PSET (U,V)
NEXT J
NEXT I
END
```

AARDE

```
REM ***een eiland op een draaiende aarde***
SCREEN 12 : CLS
WINDOW (-2,-1.5)-(2,1.5)
PI=4*ATN(1) : PHI=25 'kijkhoek
SF=SIN(PHI*PI/180) : CF=COS(PHI*PI/180)
M=6 : DIM A(M),B(M),CB(M),SB(M)
DIM U(M),V(M),U1(M),V1(M),W1(M)
REM ***data van eiland***
DATA 0,45,3,40,9,40,12,45,9,50,3,50
FOR I=1 TO M : READ A(I),B(I) : NEXT I
A(0)=A(M) : B(0)=B(M)
FOR I=0 TO M
A(I)=A(I)*PI/180 : B(I)=B(I)*PI/180
CB(I)=COS(B(I))/SQR(2) : SB(I)=SIN(B(I))
NEXT I
T=0 : H=PI/60
REM ***hoofdlus voor draaiing aarde***
DO
CIRCLE (0,0),1 : CIRCLE (0,CF),.02
FOR K=1 TO M
A=A(K) : B=B(K) : SB=SB(K) : CB=CB(K)
ST=SIN(A+T) : CT=COS(A+T)
U1(K)=(ST-CT)*CB
V1(K)=-(ST+CT)*CB*SF+SB*CF
W1(K)=(ST+CT)*CB*CF+SB*SF
NEXT K
U1(0)=U1(M) : V1(0)=V1(M) : W1(0)=W1(M)
REM ***verwijdert vorige positie***
FOR K=1 TO M
LINE (U(K-1),V(K-1))-(U(K),V(K)),0
NEXT K
REM ***brengt eiland in nieuwe positie***
FOR K=1 TO M
IF W1(K)>0 THEN LINE (U1(K-1),V1(K-1))
                  -(U1(K),V1(K))
NEXT K
FOR K=0 TO M
U(K)=U1(K) : V(K)=V1(K)
NEXT K
T=T+H
LOOP UNTIL INKEY$<>""
END
```

EILAND

```
REM ***vier kaartprojecties van een naar***
REM ***het noorden opschuivend eiland***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-2.4,-1.2)-(2.4,2.4)
M=4 : N=128
DIM A(M),B(M),X(M),Y(M),Z(M),U(M),V(M)
REM ***definitie van een vierkant gebied***
DATA -18,-18,18,-18,18,18,-18,18
FOR I=0 TO M-1 : READ A(I),B(I)
A(I)=A(I)*PI/180 : B(I)=B(I)*PI/180
NEXT I
A(M)=A(0) : B(M)=B(0)
REM ***invoer gewenste projectiemethode***
PRINT"kies projectiemethode"
PRINT"1 is equidistante cylinderprojectie"
PRINT"2 is gewone cylinderprojectie"
PRINT"3 is mercatorprojectie"
PRINT"4 is sanson-mercatorprojectie"
INPUT"keuzegetal = ",P : CLS
FOR K=0 TO 12 : VV=K*PI/30
FOR I=0 TO M-1 : FOR J=0 TO N
A=((N-J)*A(I)+J*A(I+1))/N
B=((N-J)*B(I)+J*B(I+1))/N
GOSUB projectie
SELECT CASE P
CASE 1
U=A : V=B
CASE 2
U=A : V=SIN(B)
CASE 3
U=A : V=LOG(TAN(PI/4+B/2))
CASE 4
U=A*COS(B) : V=B
END SELECT
PSET (U,V)
NEXT J : NEXT I
A$=INPUT$(1) : CLS
NEXT K : END
projectie:
X=COS(A)*COS(B) : Y=SIN(A)*COS(B)
Z=SIN(B)
XV=X*COS(VV)-Z*SIN(VV)
YV=Y : ZV=X*SIN(VV)+Z*COS(VV)
B=ATN(ZV/SQR(1-ZV*ZV+.000001))
IF XV>0 THEN A=ATN(YV/XV)
IF XV=0 THEN A=SGN(YV)*PI/2
IF XV<0 THEN A=ATN(YV/XV)+PI*SGN(YV)
RETURN
```

KAART

```
REM ***vijf projecties van een vierkant***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-3.2,-2.4)-(3.2,2.4)
N=256 : DIM X(8),Y(8),U(8),V(8)
REM ***definitie van vierkant***
H=12 'halve hoogte
VC=55 : VC=VC*PI/180 : H=H*PI/180 'centrum
X(1)=-H : Y(1)=-H : U(1)=H : V(1)=-H
X(2)=-H : Y(2)=0 : U(2)=H : V(2)=0
X(3)=-H : Y(3)=H : U(3)=H : V(3)=H
X(4)=-H : Y(4)=-H : U(4)=-H : V(4)=H
X(5)=0 : Y(5)=-H : U(5)=0 : V(5)=H
X(6)=H : Y(6)=-H : U(6)=H : V(6)=H
X(7)=-H : Y(7)=-H : U(7)=H : V(7)=H
X(8)=H : Y(8)=-H : U(8)=-H : V(8)=H
REM ***1 is equidist. cylinderprojectie***
REM ***2 is gewone cylinderprojectie***
REM ***3 is Mercatorprojectie***
REM ***4 is Sanson-Mercatorprojectie***
```

```
REM ***5 is raakvlakprojectie***
FOR P=1 TO 5
FOR K=1 TO 8 : FOR J=0 TO N
A=((N-J)*X(K)+J*U(K))/N
B=((N-J)*Y(K)+J*V(K))/N
SELECT CASE P
CASE 1
VV=VC : GOSUB projectie
U=A : V=B : U0=-1.8 : V0=-.6
CASE 2
VV=VC : GOSUB projectie
U=A : V=SIN(B) : U0=-.6 : V0=-.6
CASE 3
VV=VC : GOSUB projectie
U=A : V=LOG(TAN(PI/4+B/2)) : U0=.6 : V0=-.6
CASE 4
VV=VC : GOSUB projectie
U=A*COS(B) : V=B : U0=1.8 : V0=-.6
CASE 5
VV=0 : GOSUB projectie
U=A : V=B : U0=0 : V0=1
END SELECT
PSET (U+U0, V+V0-VV)
NEXT J : NEXT K
NEXT P : A$=INPUT$(1)
END
projectie:
X=COS(A)*COS(B) : Y=SIN(A)*COS(B)
Z=SIN(B)
XV=X*COS(VV)-Z*SIN(VV)
YV=Y : ZV=X*SIN(VV)+Z*COS(VV)
B=ATN(ZV/SQR(1-ZV*ZV+.000001))
IF XV>0 THEN A=ATN(YV/XV)
IF XV=0 THEN A=SGN(YV)*PI/2
IF XV<0 THEN A=ATN(YV/XV)+PI*SGN(YV)
RETURN
```

Hoofdstuk 12

PLANEET

```
REM ***baan van een planeet om de zon***
SCREEN 12 : CLS
WINDOW (-16,-12)-(16,12)
PI=4*ATN(1) : RANDOMIZE 11
REM ***positie zon***
LINE (-.2,0)-(.2,0) : LINE (0,-.2)-(0,.2)
REM ***baan van planeet***
E=.86
FOR J=0 TO 7
F=0 : F0=J*PI/4
DO WHILE F<2*PI
R=(1+E)/(1+E*COS(F+F0))
X=R*COS(F) : Y=R*SIN(F)
IF F=0 THEN PSET (X,Y) ELSE LINE -(X,Y)
F=F+.01*(1+E)*(1+E*COS(F))^2
LOOP
E=.95*E : A$=INPUT$(1)
NEXT J
END
```

SONDE

```
REM ***dit programma simuleert de schijn***
REM ***baar chaotische beweging van een***
REM ***ruimtesonde in de omgeving van***
REM ***twee even zware sterren. De bewe***
REM ***ging vindt plaats in een vlak.***
REM ***Ter vereenvoudiging is aangenomen***
REM ***dat de sterren stilstaan. In de***
REM ***natuur kan een dergelijke***
REM ***situatie niet voorkomen.***
REM ***De beginpositie van de satel***
REM ***liet kan in coordinaten op wille***
REM ***keurige wijze aangenomen worden***
SCREEN 12 : CLS
DEFDBL A-Z
WINDOW (-400,-300)-(400,300)
REM ***beginpositie, beginsnelheid sonde***
PRINT"geef afstand tot centrum, bv. r=80"
INPUT"r = ",R
PRINT"geef fase sonde tussen 0 en 1, bv..5"
INPUT"fase = ",PHI : PHI=8*PHI*ATN(1)
PRINT"geef grootte aanvangssnelheid, bv.13"
INPUT"snelheid = ",B
X0=R*COS(PHI) : Y0=R*SIN(PHI)
U0=-B*SIN(PHI) : V0=B*COS(PHI)
REM ***constanten***
A=20 : C=10000 : H=.004
REM initialisatie***
X=X0 : Y=Y0 : U=U0 : V=V0 : CLS
REM ***hoofdprogramma***
CIRCLE (-A,0),1,14 : CIRCLE (A,0),1,14
DO WHILE INKEY$="" AND ABS(X)<400 AND
                         ABS(Y)<300
DIST1=(X+A)*(X+A)+Y*Y
DIST2=(X-A)*(X-A)+Y*Y
C1=C*DIST1^(-1.5)
C2=C*DIST2^(-1.5)
DELU=-C1*(X+A)-C2*(X-A)
DELV=-C1*Y-C2*Y
X=X+H*U : Y=Y+H*V
U=U+H*DELU : V=V+H*DELV
PSET (X,Y),14
LOOP : BEEP : A$=INPUT$(1)
END
```

STARSHIP

```
REM ***dit programma simuleert de schijn***
REM ***baar chaotische beweging van een***
REM ***ruimtesonde in de omgeving van***
REM ***een dubbelster. De beweging***
REM ***vindt plaats in een vlak.***
REM ***De oorsprong is het zwaartepunt***
REM ***van de dubbelster***
SCREEN 12 : CLS
DEFDBL A-Z : PI=4*ATN(1)
WINDOW (-6,-4)-(10,8)
REM ***excentriciteit, positie 1ste ster***
E=.4 : X1=1/(1+E) : Y1=0
X2=-X1 : Y2=0
REM ***beginpositie, -snelheid van sonde***
R0=3 : PHI0=.001 : ABSVEL=.1 : PSI=0
X0=R0*COS(PHI0) : Y0=R0*SIN(PHI0)
U0=-ABSVEL*COS(PSI) : V0=-ABSVEL*SIN(PSI)
```

```
REM ***gravitatie constante, tijdstap***
C=.5 : H=.002
H1=H*C : H2=4*H*C*C
REM initialisatie***
X=X0 : Y=Y0 : U=U0 : V=V0 : PHI=0
REM ***hoofdprogramma***
DO WHILE INKEY$="" AND ABS(X)+ABS(Y)<16
X1V=X1 : Y1V=Y1 : X2V=X2 : Y2V=Y2
XV=X : YV=Y
R=1/(1+E*COS(PHI))
T=T+H : PHI=PHI+H1/(R*R)
X1=R*COS(PHI) : Y1=R*SIN(PHI)
X2=-X1 : Y2=-Y1
DIST1=SQR((X-X1)*(X-X1)+(Y-Y1)*(Y-Y1))
DIST2=SQR((X-X2)*(X-X2)+(Y-Y2)*(Y-Y2))
C1=H2/(DIST1*DIST1*DIST1)
C2=H2/(DIST2*DIST2*DIST2)
DELU=-C1*(X-X1)-C2*(X-X2)
DELV=-C1*(Y-Y1)-C2*(Y-Y2)
X=X+H*U : Y=Y+H*V
U=U+DELU : V=V+DELV
PSET (X1V,Y1V),0 : PSET (X2V,Y2V),0
PSET (X1,Y1),14 : PSET (X2,Y2),14
PSET(X,Y),14
LOOP : A$=INPUT$(1)
END
```

Hoofdstuk 13

KOMEET

```
REM ***kometen om de zon***
SCREEN 12 : CLS
DEFDBL A-Z
PI=4*ATN(1) : RANDOMIZE 111
PRINT"kies schaalfactor s tussen 1 en 10"
PRINT"s=1 geeft totaalbeeld zonnestelsel"
PRINT"s=10 alleen banen Jupiter en de
                    kleine planeten"
PRINT : INPUT"schaalfactor = ",S : CLS
PRINT"even wachten, er komt een komeet..."
WINDOW (-80,-60)-(80,60)
H=.15*S : DIM R(8)
DATA .4,.7,1,1.5,5.2,9.5,19,30
FOR I=1 TO 8 : READ R(I)
IF I=3 THEN COL=15 ELSE COL=4
CIRCLE (0,0),S*R(I),COL
NEXT I
DO
E=1-RND/(10*S^.3) : F=0 : F0=2*PI*RND
DO WHILE F<2*PI
R=S*50*(1-E)/(1-E*COS(F))
X=R*COS(F0+F) : Y=R*SIN(F0+F)
BL=(ABS(X)<80) AND (ABS(Y)<60)
IF BL=-1 THEN PSET (X,Y),14
IF BL=0 AND F>PI THEN GOTO komeet
F=F+H/(R*R)
IF INKEY$<>"" THEN END
LOOP
komeet:
LOOP : A$=INPUT$(1)
END
```

PLANEET1

```
REM ***zonnestelsel***
REM ***zon met acht planeten op schaal***
SCREEN 12 : CLS
PI=4*ATN(1) : RANDOMIZE TIMER
PRINT"geef schaalfactor s tussen 1 en 20"
PRINT"een grote waarde geeft de planeten
                    tot aan Mars"
INPUT"schaalfactor = ",S
INPUT"aantal jaren = ",A : CLS
WINDOW (-48,-36)-(48,36)
DIM X(8),Y(8),XV(8),YV(8)
DIM R(8),S(8),F(8),COL(8)
DATA 0,11,10,15,12,13,9,3,2
FOR I=0 TO 8 : READ COL(I) : NEXT I
DATA .387,.723,1,1.524,5.2,9.54,19.18,30.06
FOR I=1 TO 8
READ R(I) : S(I)=R(I)^(-1.5)
NEXT I
FOR I=1 TO 8 : F(I)=2*PI*RND : NEXT I
H=.01 : T=0
DO WHILE INKEY$="" AND T<2*PI*A
CIRCLE (0,0),S*.1,15 : PAINT (0,0),15
FOR J=1 TO 8
F=F(J)+S(J)*T
X(J)=S*R(J)*COS(F) : Y(J)=S*R(J)*SIN(F)
IF T>0 THEN CIRCLE (XV(J),YV(J)),.3,0
IF T>0 THEN PAINT (XV(J),YV(J)),0
CIRCLE (X(J),Y(J)),.3,COL(J)
PAINT (X(J),Y(J)),COL(J)
XV(J)=X(J) : YV(J)=Y(J)
T=T+H
NEXT J
LOOP : A$=INPUT$(1)
END
```

PLANEET2

```
REM ***schijnbare planetenbeweging***
REM ***aarde met een planeet***
SCREEN 12 : CLS
WINDOW (-6,-4.5)-(6,4.5)
REM ***straal en inclinatie baan planeet***
R=1.5 : PHI=.3
R1=R*COS(PHI) : R2=R : R3=R*SIN(PHI)
H=.005 : C=R^(-1.5) : T=0
CIRCLE (0,0),.1,15 : PAINT (0,0),15 'zon
DO WHILE T<50 AND INKEY$=""
X=COS(T) : Y=SIN(T)
X1=R1*COS(C*T) : Y1=R2*SIN(C*T)
Z1=-R3*COS(C*T)
XV=COS(T-H) : YV=SIN(T-H)
X1V=R1*COS(C*(T-H)) : Y1V=R2*SIN(C*(T-H))
CIRCLE (XV,YV),.08,0 : PAINT (XV,YV),0
CIRCLE (X,Y),.06,14 : PAINT (X,Y),14 'aarde
CIRCLE (X1V,Y1V),.08,0 : PAINT (X1V,Y1V),0
CIRCLE (X1,Y1),.06,12 : PAINT (X1,Y1),12
S=SQR((X-X1)*(X-X1)+(Y-Y1)*(Y-Y1)+Z1*Z1)
U=3*(X1-X)/S : V=3+Z1/S
PSET (U,V) : T=T+H
LOOP : A$=INPUT$(1)
END
```

Hoofdstuk 14

HARM

```
REM ***harmonisch beweging***
REM ***numerieke integratie van de***
REM ***bewegingsvergelijkingen***
SCREEN 12 : CLS
WINDOW (0,-3)-(32,3)
LINE (0,0)-(32,0) : LINE (0,2)-(0,-2)
PRINT"neem demping tussen 0 and 2"
INPUT"dempingsconstante = ",MU
LOCATE 1,1 : PRINT SPACE$(80)
H=.002 : U=2 : V=0 : T=0 'start
DO WHILE T<32 AND INKEY$=""
W=U : U=U+H*V
V=V-H*(U+MU*V)
T=T+H
PSET (T,U)
LOOP : A$=INPUT$(1)
END
```

POL1

```
REM ***de vergelijking van Van der Pol***
SCREEN 12 : CLS
H=.001 : TMAX=50
WINDOW (0,-3)-(TMAX,3)
LINE (0,0)-(TMAX,0) : LINE (0,2)-(0,-2)
PRINT"kies relaxatiefactor tussen 0 en 4"
INPUT"relaxatie = ",MU
LOCATE 1,1 : PRINT SPACE$(80)
U=.1 : V=0 : T=0 'start
DO WHILE T<TMAX AND INKEY$=""
W=U : U=U+H*V
V=V-H*(W+MU*(W*W-1)*V)
T=T+H
PSET (T,U)
LOOP
END
```

POL2

```
REM ***de vergelijking van Van der Pol****
REM ***fase diagram***
SCREEN 12 : CLS
PRINT"kies relaxatiefactor tussen .2 en 1"
INPUT"relaxatie = ",MU : CLS
WINDOW (-4,-3)-(4,3)
LINE (-4,0)-(4,0) : LINE (0,-3)-(0,3)
H=.002 : KMAX=20000
U=0 : V=.1
FOR K=1 TO KMAX
W=U : U=U+H*V
V=V-H*(W+MU*(W*W-1)*V)
PSET (U,V)
NEXT K
END
```

POL3

```
REM ***een model van een horloge****
SCREEN 12 : CLS
DEF FNF(U)
FNF=.2*(U*U-.2)*(U*U-2)
END DEF
PRINT"start de klok voor s>.9"
INPUT"stoot s = ",S
WINDOW (-8,-6)-(8,6)
LINE (-4,0)-(4,0) : LINE (0,-3)-(0,3)
H=.05 : KMAX=20000
U=S : V=0
FOR K=1 TO KMAX
IF INKEY$<>"" THEN END
W=U : U=U+H*V
V=V-H*(W+V*FNF(W))
PSET (U,V)
NEXT K
END
```

Hoofdstuk 15

TORUS

```
REM ***torus oppervlak***
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3)
PI=4*ATN(1) : ALFA=2.1 'kijkhoek
SA=SIN(ALFA) : CA=COS(ALFA) :
TA=TAN(ALFA)
A=2.5 : B=.8 : H=.003 : T=0
LOCATE 1,1
PRINT"indien klaar, druk op een toets"
DO
U=11*T
X=(A+B*COS(U))*COS(T)
Y=(A+B*COS(U))*SIN(T)
Z=B*SIN(U)
GOSUB graphics
T=T+H
LOOP UNTIL INKEY$<>""
GOSUB toruslines
LOCATE 1,1 : PRINT SPACE$(80)
A$=INPUT$(1)
END
graphics:
W1=SA*SIN(T)*COS(U)+CA*SIN(U)
IF W1>0 THEN COL=14
IF W1<=0 AND W>=0 THEN COUNT=0
IF W1<=0 AND W<0 THEN
COUNT=COUNT+1
IF COUNT MOD 8>4 THEN COL=14 ELSE COL=0
END IF
W=W1
PSET (X,CA*Y-SA*Z),COL
RETURN
toruslines:
FOR T=0 TO 2*PI STEP .005
U1=-ATN(TA*SIN(T)) : U2=U1+PI
X1=(A+B*COS(U1))*COS(T)
X2=(A+B*COS(U2))*COS(T)
Z1=CA*(A+B*COS(U1))*SIN(T)-SA*B*SIN(U1)
Z2=CA*(A+B*COS(U2))*SIN(T)-SA*B*SIN(U2)
```

```
PSET (X1,Z1)
PSET (X2,Z2)
NEXT T
RETURN
circles:
FOR T=0 TO 2*PI STEP .05
FOR U=0 TO 2*PI STEP .05
X0=(A+B*COS(U))*COS(T)
Y0=(A+B*COS(U))*SIN(T)
Z0=B*SIN(U)
PSET (X0,CA*Y0-SA*Z0),14
NEXT U : NEXT T
RETURN
```

KLOK

```
REM ***model van een stabiele klok***
SCREEN 12 : CLS
WINDOW (-2,-1.5)-(2,1.5) : PI=4*ATN(1)
PRINT"kies de demping tussen 0 en 2"
INPUT"dempingsfactor = ",MU
PRINT"kies de storing tussen .5 en 1.5
                    maar niet = 1"
INPUT"storing = ",A : CLS
IF A=1 THEN END
LINE (-1.5,0)-(1.5,0)
LINE (0,-1.5)-(0,1.5)
C=1/((A-1)*(A-1))-1
IF A>1 THEN F=0 : ELSE F=PI
PSET (-1+A,0)
FOR K=0 TO 500 : T=K/10
R=SQR(1/(1+C*EXP(-2*MU*T)))
X=R*COS(T+F) : Y=R*SIN(T+F)
LINE -(X,Y)
NEXT K : A$=INPUT$(1)
END
```

KLOKM

```
REM ***model van een hartklok***
REM ***volgens een experiment door Mines***
N=500 'neem N groter/kleiner om het programma
                    te vertragen/versnellen
SCREEN 12 : CLS
WINDOW (-4,-3)-(4,3) : PI=4*ATN(1)
DEF FNA(T,A)
SELECT CASE A
CASE <1
FNA=T-ATN(A*SIN(T)/(1+A*COS(T)))
CASE >1
FNA=ATN(SIN(T)/(A+COS(T)))
CASE 1
FNA=T/2-PI*INT((PI+T)/(2*PI))
END SELECT
END DEF
DO : T=0 : LOCATE 1,1
PRINT SPACE$(160) : LOCATE 1,1
PRINT"kies een stimulus tussen .5 en 2
                    (de waarde 1 is gevaarlijk)"
PRINT"geef stimulus of druk op 'Enter' om
                    te stoppen"
INPUT"stimulus = ",A
IF A=0 THEN END ELSE CLS
PRINT"druk op een willekeurige toets voor
                    een stimulus"
PRINT"of druk op Q voor herhaling"
LINE (-2,0)-(2,0),8 : LINE (0,-2)-(0,2),8
MU=2 : H=.05 'tijdstap
T0=0 : U0=0 : C=0
DO
CIRCLE (X,Y),.01,8 : PAINT(X,Y),8,8
U=T-T0+U0
R=1/SQR(1+C*EXP(-2*MU*(T-T0)))
X=R*COS(U) : Y=R*SIN(U)
CIRCLE (X,Y),.01,15 : PAINT (X,Y),15,15
A$=INKEY$
IF A$="Q" OR A$="q" THEN GOTO herhaal
IF A$<>"" THEN GOSUB stoot
IF A$="Q" THEN GOTO herhaal
T=T+H
FOR I=0 TO N : NEXT I
LOOP
herhaal:
LOOP
END
stoot:
T0=T : R0=SQR(1+2*A*COS(U)+A*A)
IF R0<.2 THEN GOSUB txt : A$="Q"
C=1/(R0*R0)-1 : U0=FNA(U,A)
RETURN
txt:
LOCATE 1,1 : PRINT SPACE$(150) : LOCATE 1,1
PRINT"fatale stoot"
PRINT"druk op een toets voor herhaling"
B$=INPUT$(1)
RETURN
```

CIRKELH1

```
REM ***model x'=c+x-bsin(2πx) mod 1 ***
REM ***x'x diagram voor vaste a en c***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-2,-1)-(6,5)
A=.9 : B=A/(2*PI) : C=.4 'data
LINE (0,4)-(0,0) : LINE -(4,0)
LINE (0,0)-(4,4)
FOR K=0 TO 400 : X=4*K/400
Y=C+X-B*SIN(2*PI*X)
IF Y<4 THEN PSET (X,Y)
NEXT K
REM ***hoofdlus***
X=.2 : 'start
FOR N=1 TO 100
Y=C+X-B*SIN(2*PI*X)
IF Y<4 THEN LINE (X,X)-(X,Y) : LINE -(Y,Y)
X=Y
NEXT N : A$=INPUT$(1)
END
```

CIRKELH2

```
REM ***model x'=c+x-bsin(2πx) mod 1 ***
REM ***x'x diagram voor vaste a en c***
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (-.5,-.3)-(1.5,1.2)
A=.9 : B=A/(2*PI) : C=.4 'data
P$="##.### "
LINE (0,0)-(1,1),,B
LINE (0,0)-(1,1)
```

```
LOCATE 1,1 : PRINT USING P$;A,C
FOR K=0 TO 400 : X=K/400
Y=C+X-B*SIN(2*PI*X) : Y=Y-INT(Y)
PSET (X,Y)
NEXT K
REM ***hoofdlus***
X=.2 : 'start
FOR N=1 TO 100
Y=C+X-B*SIN(2*PI*X) : Y=Y-INT(Y)
LINE (X,X)-(X,Y) : LINE -(Y,Y)
X=Y
NEXT N : A$=INPUT$(1)
END
```

CIRKELH4

```
REM ***model x'=c+x-b sin(2πx) ***
REM ***diagram van periode voor vaste a***
REM ***alleen periode <30 is afgebeeld***
DEFDBL A-Z
SCREEN 12 : CLS : PI=4*ATN(1)
WINDOW (0,-10)-(639,40)
LINE (120,0)-(120,30)
FOR K=0 TO 6
LINE (110,5*K)-(120,5*K) : NEXT K
FOR K=0 TO 30
LINE (115,K)-(120,K) : NEXT K
FOR K=0 TO 10
LINE (120+40*K,0)-(120+40*K,-1) : NEXT K
FOR K=0 TO 100
LINE (120+4*K,0)-(120+4*K,-.5) : NEXT K
A=1 : B=A/(2*PI) 'vaste waarde van a
DEF FNA(X)=X-B*SIN(2*PI*X)
FOR M=0 TO 400 : C=M/400
IF INKEY$<>"" THEN END
X=.2 'start
FOR N=1 TO 1000
X=C+FNA(X)
NEXT N
X0=X-INT(X) : X=X0FOR K=1 TO 99
Y=C+FNA(X) : Y1=Y-INT(Y)
IF ABS(Y1-X0)<.0001 THEN
Q=K : P=INT(Y) : EXIT FOR
ELSE
Q=999
END IF
X=Y
NEXT K
LOCATE 1,1 : PRINT SPACE$(18)
LOCATE 1,1 : PRINT P,Q
IF Q<30 THEN LINE (120+M,0)-(120+M,Q)
NEXT M : A$=INPUT$(1)
END
```

CIRKELH5

```
REM ***model x'=c+x-asin(2πx)/(2π) ***
REM ***periode voor 0<c<1 en 0<a<2.5 ***
SCREEN 12 : CLS
DEFDBL A-Z
XM=320 : YM=400
EPS=.000001 : PI=4*ATN(1)
N1=200 : N2=400
FOR J=1 TO N2
FOR I=0 TO N1
IF INKEY$<>"" THEN END
B=J/N2 : C=.5+I/(2*N1)
X=.2
FOR N=0 TO 400
X=C+X-B*SIN(2*PI*X)
NEXT N
X0=X-INT(X) : X=X0
FOR K=1 TO 27
Y=C+X-B*SIN(2*PI*X) : Y1=Y-INT(Y)
IF ABS(Y1-X0)<EPS THEN
COL=1+K MOD 14 : GOTO graphics
END IF
X=Y
NEXT K
COL=1
graphics:
PSET (XM+I,YM-J),COL : PSET (XM-I,YM-J),COL
NEXT I
NEXT J : A$=INPUT$(1)
END
```

Hoofdstuk 16

BAKERB

```
REM ***de transformatie van de bakker***
REM ***transformaties van een blok***
SCREEN 12 : CLS
WINDOW (-.5,-.3)-(1.5,1.2)
A=.2 : B=.2 'afmetingen van inwendig blok
PMAX=6 'maximale aantal iteratiestappen
FOR P=1 TO 6
LINE (0,0)-(1,1),,B
LINE (.5-A,.5-B)-(.5+A,.5+B),14,B
KMAX=2^(P+6) : LMAX=2^(8-P)
FOR K=0 TO KMAX : FOR L=0 TO LMAX
X=.5-A+2*A*K/KMAX : Y=.5-B+2*B*L/LMAX
FOR N=1 TO P
IF N=P THEN PSET (X,Y),14
GOSUB trafo
NEXT N
NEXT L : NEXT K : A$=INPUT$(1) : CLS
NEXT P : A$=INPUT$(1)
END
trafo:
IF X<.5 THEN X=2*X : Y=Y/2
ELSE
X=2*X-1 : Y=(Y+1)/2
END IF
RETURN
```

BAKERP

```
REM ***de transformatie van de bakker***
REM ***de chaotische baan van een punt***
SCREEN 12 : CLS
WINDOW (-.5,-.3)-(1.5,1.2)
DEFDBL A,X,Y
A=2^(-20) : RANDOMIZE 11
X=.1 : Y=.2
LINE (0,0)-(1,1),,B
FOR N=1 TO 10000
```

```
IF N>40 THEN PSET (X,Y)
GOSUB trafo
NEXT N : A$=INPUT$(1)
END
trafo:
IF X<.5 THEN Y=Y/2 ELSE Y=(Y+1)/2
X=2*X+A*RND : X=X-INT(X)
RETURN
```

LYAPUN1

```
REM ***grafiek gemiddelde groeisnelheid***
REM ***van x=ax(1-x) ***
SCREEN 12 : CLS
DEFDBL A-Z
WINDOW (0,-1)-(639,3)
REM ***interval van de parameter a"
AMIN=1 : AMAX=4
XM=320 : N=200
REM ***hoofdprogramma***
FOR I=0 TO 2*N
IF INKEY$<>"" THEN
ENDA=AMIN+I*(AMAX-AMIN)/(2*N)
END IF
REM ***beginbaan***
X=.51
FOR J=1 TO 100
X=A*X*(1-X)
NEXT J
REM ***voortgezette baan***
S=0
FOR K=1 TO 1000
X=A*X*(1-X) : T=A*(1-2*X)+.00001
S=S+LOG(ABS(T))
NEXT K
W=EXP(S/1000)
LINE (XM-N+I,0)-(XM-N+I,W)
NEXT I : A$=INPUT$(1)
END
```

LYAPUN2

```
REM ***grafiek van de Lyapunov getallen***
REM ***van x=ax(1-x) ***
SCREEN 12 : CLS
DEFDBL A-Z
WINDOW (0,-2)-(639,2)
REM ***interval van de parameter a***
AMIN=3 : AMAX=4
XM=320 : N=270
REM ***hoofdprogramma***
FOR I=0 TO 2*N
IF INKEY$<>"" THEN END
A=AMIN+I*(AMAX-AMIN)/(2*N)
X=.51 : LMAX=2000
REM ***beginbaan***
FOR K=1 TO 100
X=A*X*(1-X)
NEXT K : S=0
REM ***voortgezette baan***
FOR L=1 TO LMAX
X1=A*X*(1-X)
S=S+LOG(ABS(1-2*X)+.000001)
X=X1
NEXT L
S=LOG(A)+S/LMAX
LINE (XM-N+I,0)-(XM-N+I,S)
NEXT I : A$=INPUT$(1)
END
```

LYAPUN3

```
REM ***grafiek Lyapunov getallen van het***
REM ***gemengde systeem x'=ax(1-x) ***
REM ***afgewisseld door x'=bx(1-x) ***
SCREEN 12 : CLS
DEFDBL A-Z
WINDOW (0,-1)-(639,1)
REM ***interval van de parameter a***
AMIN=3.5 : AMAX=4
B=3 'vaste waarde van b
XM=320 : N=270
REM ***hoofdprogramma***
FOR I=0 TO 2*N
IF INKEY$<>"" THEN END
A=AMIN+I*(AMAX-AMIN)/(2*N)
X=.5 : LMAX=2000
REM ***beginbaan***
FOR K=1 TO 100
X1=A*X*(1-X) : X=B*X1*(1-X1)
NEXT K : S=0
REM ***voortgezette baan***
FOR L=1 TO LMAX
X1=A*X*(1-X) : X2=B*X1*(1-X1)
P=ABS((1-2*X1)*(1-2*X2))
S=S+LOG(P+.00000001)
X=X2
NEXT L
S=(S/LMAX+LOG(A*B))/2
LINE (XM-N+I,0)-(XM-N+I,S)
NEXT I : A$=INPUT$(1)
END
```

LYAPUN4

```
REM ***diagram van de negatieve Lyapunov***
REM ***exponenten, type aabab***
DEFDBL A-Z
SCREEN 12 : CLS
XM=320 : YM=250
REM ***beeldbegrenzing***
AMIN=3 : AMAX=4 : BMIN=3 : BMAX=4
N1=280 : N2=INT(.75*N1) : LMAX=500
REM ***hoofdprogramma***
FOR I=-N1 TO N1 : FOR J=-N2 TO N2
A=AMIN+(N1+I)*(AMAX-AMIN)/(2*N1)
B=BMIN+(N2+J)*(BMAX-BMIN)/(2*N2)
C=A*A*A*B*B
IF INKEY$<>"" THEN END
X=.51
FOR K=1 TO 100
X1=A*X*(1-X) : X2=A*X1*(1-X1)
X3=B*X2*(1-X2) : X4=A*X3*(1-X3)
X5=B*X4*(1-X4)
IF ABS(X5)>2 THEN COL=0 : GOTO graphics
X=X5
NEXT K
S=0
```

```
FOR L=1 TO LMAX
X1=A*X*(1-X) : X2=A*X1*(1-X1)
X3=B*X2*(1-X2) : X4=A*X3*(1-X3)
X5=B*X4*(1-X4)
T=(1-2*X1)*(1-2*X2)*(1-2*X3)*(1-2*X4)
                        *(1-2*X5)
IF ABS(X5)>2 THEN COL=0 : GOTO graphics
S=S+LOG(ABS(T)+1E-10)
X=X5
NEXT L
S=-(LOG(C)+S/LMAX)  'lyap. exponent is s/5
REM ***kleurencode***
IF S>0 THEN COL=2+INT(8*S) MOD 14
ELSE
COL=1
END IF
graphics:
PSET (XM+I,YM-J),COL
NEXT J : NEXT I
END
```

Bibliografie

M.F. Barnsley, *Fractals Everywhere*. Academic Press, 1988

H.W. Broer en F. Verhulst, *Dynamische systemen en chaos. Een revolutie vanuit de wiskunde*. Epsilon Uitgaven, Utrecht, 1990

J. Gleick, *Chaos. De derde wetenschappelijke revolutie*. Contact, 1989

Hans Lauwerier, *Fractals. Meetkundige figuren in eindeloze herhaling*. Aramith Uitgevers, Bloemendaal, 5de druk, 1992

Hans Lauwerier, *Een wereld van fractals*. Aramith Uitgevers, Bloemendaal, 1990

H.A. Lauwerier, *Modellen met de microcomputer. Experimentele wiskunde*. Epsilon Uitgaven, Utrecht, 1989

B.B. Mandelbrot, *The Fractal Geometry of Nature*. Freeman and Co., 1977

Natuur & Techniek. Natuurwetenschappelijk en technisch maandblad. Centrale Uitgeverij en Adviesbureau

P. Prusinkiewicz en J. Hanan, *Lindenmayer Systems. Fractals and Plants*. Springer Verlag, 1989

P. Prusinkiewicz en A. Lindenmayer, *The Algorithmic Beauty of Plants*. Springer Verlag, 1990

Mensen zijn nieuwsgierig en willen graag experimenteren. Soms voeren ze allerlei experimenten in huis uit met het gevaar van elektrische schokken en chemische explosies. Tegenwoordig beschikken velen thuis over een computer waarmee ze, misschien zonder het nog te weten, een heel laboratorium in huis hebben gehaald. Het is nu mogelijk vele verschijnselen van de natuur te begrijpen aan de hand van computerexperimenten.
In dit boek beschrijft de auteur een groot aantal computerexperimenten waarmee men inzicht kan krijgen in zulke uiteenlopende zaken als het verloop van een epidemie, de groei van planten, de baan van een ruimtevaartuig, milieuvervuiling en de werking van een pacemaker. Men kan zijn eigen planetarium maken, en men kan gaan begrijpen hoe bedrieglijk een Mercatorkaart kan zijn. Dat alles met eenvoudige formules, die niet meer dan een minimum aan wiskundige kennis veronderstellen.
Alle experimenten zijn uitvoerbaar met eenvoudige computerprogramma's, geschreven in de moderne en goed leesbare QBasic vorm. De stof is voor een deel ontleend aan de maandelijkse bijdragen van de auteur in het tijdschrift Natuur en Techniek.
Hans Lauwerier is bekend van boeken als **'Fractals'**, **'Symmetrie'**en **'Oneindigheid'**.

ISBN 90-6834-106-5

9 789068 341065

ARAMITH UITGEVERS